An ADER-WENO Approach to Modelling Low Mach Number Multiphase Flows

For Granular Materials in Moving Geometries

B.A. Young

Center for Scientific Computing
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy

Downing College August 2017

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others.

Word Count: 14,055 (excluding tables, figure legends and bibliography)

B.A. Young
August 2017

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Nikiforakis as well as Dr Michael for all the help and guidance given to me throughout the project. Additionally I would like to thank Aoife Hannon for the many fruitful discussions had during the course of the work.

For my friends and family

Abstract

This work presents a novel, high order numerical algorithm for solving multiphase granular flows within two dimensional arbitrary moving geometries. This took the form of an ADER-WENO, non-conservative solver for the Baer-Nunziato equations combined with a ghost-fluid type method for solid boundaries, written specifically for this work. The scheme was successfully validated against a wide variety of test cases, including experimental results. Finally, the solver was used to solve two novel demonstration problems, for which no numerical result had previously been produced.

Keywords: ADER-WENO; Baer-Nunziato; Granular Flow; Non-Conservative Hyperbolic Schemes; Ghost-Fluid Method; Moving Solid Geometries.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ALE	Arbitrary Lagrangian-Eulerian
BN	Baer-Nunziato
BNRP	Baer-Nunziato Riemann Problem
CFL	Courant-Freidrich-Lewy
DG	Discontinuous Galerkin
EOS	Equation of State
FV	Finite Volume
GFM	Original Ghost fluid method for multiphase flows
MGFM	Modified Ghost Fluid Method for solid boundaries
MPF	Multiphase flow
PDE	Partial Differential Equation
RP	Riemann Problem
WENO	Weighted Essentially Non-Oscillatory

Constants

$\bar{\pi}$	Stiffened Pressure
δ	Interphase drag parameter
γ	Ratio of Specific Heats
μ_c	Compaction relaxation parameter
b	Vitrial Equation of State parameter
q	Heat release parameter
N	Scheme Order

Variables

α_k	Volume fraction of phase k
β	Compaction tensor
$\hat{\mathbf{N}}$	Unit normal vector
ϕ	Ghost Fluid Level Set function

ρ_k	Density of phase k
$\mathbf{U}$	Vector of conserved variables
$\mathbf{w}$	Vector of primitive variables
a_k	Speed of sound in phase k
B	Compaction Potential Energy
c_k	Speed of sound in phase k
p_k	Partial pressure of phase k
u_k	Horizontal velocity of phase k
v_k	Vertical velocity of phase k
x'	Dimensional quantity x

1 Introduction

The purpose of this study was to develop a numerical solver capable of accurately simulating two dimensional multiphase granular flows within moving complex geometries. Initially, a literature review was used to analyze the physics and mathematics of multiphase granular flows, with a strong emphasis placed on relevant numerical considerations, such as implementation of boundary conditions and stability. This resulted in the development and implementation of a primitive variable ADER-WENO scheme for the multiphase Baer-Nunziato (BN) model, combined with a modified ghost fluid method (MGFM) for simulating complex solid boundaries. The developed scheme was numerically, experimentally and theoretically validated, before being used to simulate several novel problems that demonstrate its capabilities.

1.1 Background

Multiphase fluid mechanics is the study of systems involving two or more interacting phases of matter. These systems are ubiquitous throughout both the natural and industrial world and come in such diverse forms as: sediment transport in rivers; gas flow in combustion engines; blood flow; avalanche mechanics and galaxy formation. The ability to accurately model and predict these phenomena is therefore highly important for future developments in science, technology and medicine. One of the most interesting developments in the field within the last 20 years has been the application of multiphase flow models to dense granular and particulate systems, particularly as the large number of particles within these systems, $\approx 10^9$ per m^3, elude traditional discrete models. More often than not, this physical and mathematical complexity mean that analytical models are impossible to derive or may be far too over simplified to accurately model the underlying phenomena. This is especially true in the case of flows with complex geometries, non-negligible surface tension and non-ideal equations of state.

Despite the pervasiveness of these flows, applicable numerical methods are still a recent and developing area of study with many unsolved problems within the field [6, 7, 8]. One particular area of interest, discussed within this work, is the study of numerical models for multiphase flows in moving geometries. Udaykamar et al. [9] presented one of the original methods for modeling MPF interaction with complex, moving geometries in the form of their ELAFINT code. This used an Eulerian-Lagrangian (ALE) formulation for dealing with the phase interaction and a cut-cell method to model moving geometries. Whilst this can be used to accurately model imiscible fluid interaction, it performed poorly in dealing with granular and mixed phases. This is because the ALE method does not model sub-cell interactions between the two phases, as each computational cell is only able to contain one distinct phase, with phase-phase interactions emulated using movements within the mesh.

Hu et al.'s ALE model [10] improved upon this by including an additional equation describing phase interactions at the sub-grid level, thus allowing some modeling of mixing. However, reliance on the ALE formulation for two distinct phases means that porous granular flow models require each individual particle to be resolved, thus making it impractical to model large scale mass flows. The ALE solver does allow for denser granular flows to be modeled, but only if the effects of compaction and dispersion are ignored. As an alternative to continuum type models, there has also been significant interest in modeling granular type flows using discrete element and particle based methods [11].

This demonstrates a significant gap within current literature. Whilst discrete particle element methods may work well for sparse granular flows and ALE models for dense, non-compacting flows, there is little existing work on modeling the interaction between dense compacting granular flows and moving geometries. Thus the aim of this project was to develop a multiphase solver capable of simulating multiphase granular flows in moving geometries within a framework for a generic mixed Mach number, continuum model based multiphase flow.

The introduction is structured as follows, in Section 1.2, the physics of granular flows is discussed with a particular emphasis on the physical properties required to be predicted in the numerical model. These include phenomena such as compaction, dispersion and shock wave generation. Then Sections 1.3-1.6 contain a literature review, exploring the wealth of potential models for multiphase flows, algorithms for dealing with moving solid boundaries and the numerical methods available for solving these models.

Following the literature review, the chosen Baer-Nunziato (BN) model is discussed in further detail in Section 3 along with descriptions of the additional equation of state and phase interaction required to fully simulate the variety of multiphase flows encountered. In Section 4, the high resolution ADER-WENO method and solution algorithm the BN equations is introduced, before the total algorithm is validated using a variety of test cases in Section 5. In Section 6, the validated solver is them used on several example problems to demonstrate its novel capabilities before the complete work is discussed in detail within Section 7.

1.2 The Physics of Granular, Compacting Flows

Compaction is the process through which a granular material such as; sand, soil, or powder locally increases its volume fraction and decreases it porosity due to pressure or velocity gradients within the system. The simplest example of this is the effect produced by pushing a piston through a mixture of sand and air. The sand adjacent to the moving piston wall will compact and reduce the space in between each grain, compressing or displacing any air present. The compaction process can be seen clearly in Figure 1.

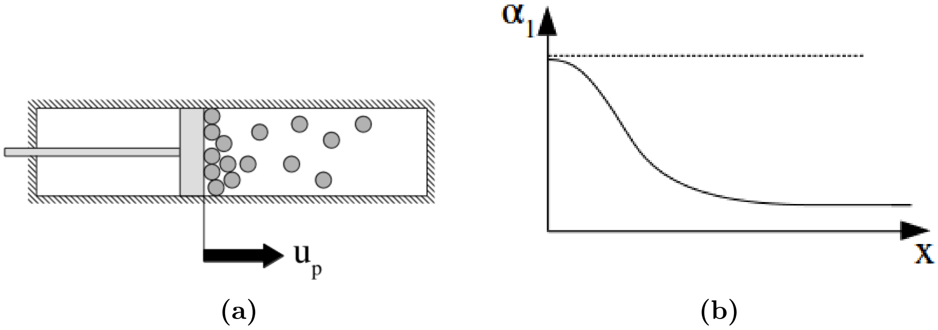


Figure 1: Diagram of (a) compaction of grains due to displacement of a piston, (b) the corresponding increase in granular volume fraction, the dotted line shows a theoretical maximum where grains have perfect packing

Additionally, granular materials can undergo the opposite process, known as dispersion or cavitation, in which the flow acts to locally decrease the solid volume fraction to avoid compression of the solid grains. An example of this would be moving the piston in Figure 1 leftward - the grains would separate and allow a larger volume of air to fill the space between particles. Although granular materials consist of many individual solid (incompressible) grains, they are often modeled as a compressible fluid, owing to their shared properties, such as: developing shock waves; flowing to reduce pressure gradients; and inducing drag around objects [12]. A comprehensive review of numerical methods for granular flow modeling can be found in [13, 14].

One of the most prominent contimuum based models of granular flow (and the model used here) was first proposed by Baer and Nunziato [15] as a multiphase granular detonation model, but has since developed into a more general framework. With additional source terms, the Baer-Nunziato equations can also be used to model compaction/dispersion phenomena, for example through the additions outlined in Schwendeman et al. [5]. These additions come in the form of closure and source terms to the Baer-Nunziato equations, (although they can also be applied in the context of other multiphase flow formulations). These terms drive the two phases towards a thermodynamic equilibrium through an exchange of momentum, density, pressure and volume fraction, which physically manifests itself as the processes of compaction, cavitation, drag, heat transfer and surface tension.

The Two Dimensional Extension

The natural extension to the 1D piston compaction problem is to examine how a granular material will behave when compacted by a two dimensional object such a cylinder or a rectangular prism. Figure 2 shows one such example - an experimental image of the granular flow around a cylinder. It should be noted that this experiment was run at a much lower Mach number than the problems modeled within this work and thus is only here as an example of the type of physical phenomena produced by these flows.

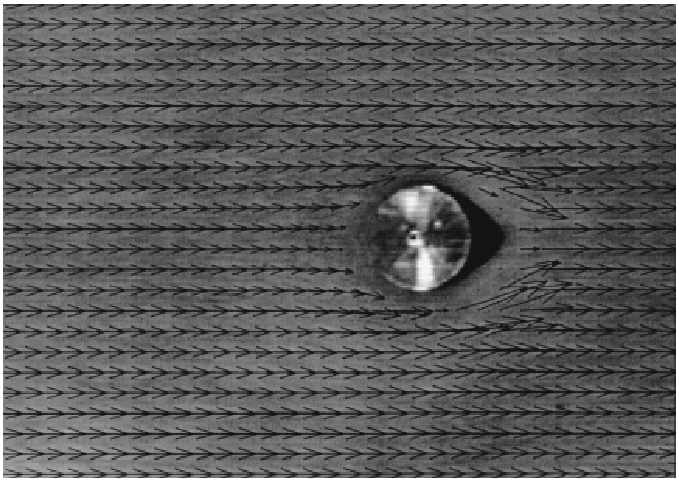


Figure 2: Experimental image of air/sand flow around a cylinder with superimposed velocity profile. Cavitation can be seen at the aft of the cylinder, whilst the darker region at the fore indicates compaction taking place. Image taken from FIG.11 of [1].

Moving into 2D means that several of the granular flow phenomena can be observed within the same example problem, these include: acceleration of the flow around the top and bottom of the cylinder; cavitation at the aft and compaction at the fore. Whilst these flows are more interesting and relevant to physical problems, the additional geometric consideration make modeling significantly harder than the 1D equivalent. The curved geometry needs to be applied as an additional solid boundary, that may also need to be tracked if the solid object is accelerating, changing direction or moving relative to the boundaries of the flow.

To accurately capture the physics of these types of problems, a complete numerical algorithm will, therefore, require two underlying parts: a model of the multiphase flow and a method for applying solid, moving boundary conditions. The following sections contain a review of both of these types of model.

1.3 Review of Multiphase Models

Whilst there are a huge variety of methodologies available for modeling multiphase granular flows, including discrete element and particle based methods, the approach taken in this work was to use a continuum based model. The reasoning behind this was due to the wealth of existing research within continuum model based approaches as well as the large number of validation cases that exist - meaning accurately verifying the numerical model was significantly easier than more novel approaches.

In this section, some of the most common approaches to modeling multiphase flows are reviewed, including the Baer-Nunziato model used in the later work within this paper.

Arbitrary Lagrangian-Eulerian methods

In this formulation, each phase is modeled using the single phase Euler equations cast in Lagrangian form, i.e in the frame of reference of each element. The elements then move in accordance with the movement of the fluid. This allows for incredibly accurate tracking of the material interface, as the interface of the two phases will always lie on an interface between two computational elements. However this accuracy comes at a cost. The ALE scheme cannot be used for miscible fluids and in the case of bubbles/grains would require very small computational elements, adding significant computational cost. Additionally any turbulence or vorticity within the fluid would produce a very skewed mesh, greatly reducing accuracy or requiring computationally expensive re-meshing every few time steps. [16, 17]

Level-set methods

Level set methods (including the original Ghost fluid method) work by defining a signed level set function, positive within phase 1, negative within phase 2 with magnitude equal to the shortest distance to the phase/phase interface. The two phases are solved separately on two separate meshes at each time step, before the two solutions are 'stitched together' by extrapolating variables conserved across the material interface (such as modified pressure, velocity and entropy) to effectively form a new set of boundary conditions under which the two systems are solved. This is explained in greater depth in Section 3.7. This has similar advantages as the ALE formulation discussed in the previous section: good interface tracking; no mixing of phases; generic equations of state etc, but without the additional complications introduced with the Lagrangian mesh. These reasons also make it particularly suited for modeling moving fluid/solid interfaces without using an unstructured mesh scheme. [18, 19]

Volume of Fluid Methods

The volume of fluid method is similar to the level set method described in the previous section in that an additional function, $C(x)$, is defined which characterizes the presence of each phase. $C(x)$ takes the form of a binary function ($C(x) = 1$ in phase 1 and $C(x) = 0$ in phase 2), often defined on a higher resolution grid than the 2 phases (owing to the lower memory requirements). At each time step, an algorithm such as the 'donor-acceptor scheme' or the 'compressive interface capturing scheme' uses the velocity, density and pressure of each phase to update $C(x)$, thereby moving the interface between the phases. These algorithms allow the scheme to conserve the total mass of each phase, whilst maintaining a high resolution interface. This method was originally derived for single phase free surface problems (ignoring surface tension and inter-phase effects), thus making the model less physical than the others discussed here. [20]

The Diffuse Interface Method

The Diffuse Interface Method combines a single phase Euler equation with an additional advection equation describing the volume fraction at each point within the computational domain. Generally this means that the location of the phase interface is not as accurately tracked as the other methods (although high resolution shock-capturing methods resolve this), resulting in some mixing of the phases. The two equations of state describing each phase also need to be combined in the form of a mixing rule to describe the generalized system. This is difficult to implement within a generalized multiphase solver, but allows standard methods to be used to solve the combined Euler/interface equation.

To eliminate the need for a set of mixing rules, the diffuse interface method can be further generalized to a set of two conservation laws (one for each phase) combined with an equation describing the evolution of the volume fraction. This is the basis for the Baer-Nunziato model of multiphase flow. [21]

The Baer-Nunziato Model

The Baer-Nunziato model is essentially two sets of compressible Euler equations combined with an additional equation describing the volume fraction ratio of each phase within the domain. This formulation has several advantages over the other methods discussed here. Firstly, no complicated moving mesh is required to simulate the interaction between phases. This also means that the two phases are able to mix and can therefore compact or disperse. Additionally the Baer-Nunziato model is a hyperbolic PDE, similar to the Euler equation and thus standard numerical methods can be used. There are, however some disadvantages. The B-N equations require a high resolution method in order to accurately capture the discontinuities in volume fraction between the two phases which incurs significant computational cost. Failure to do so can lead to instabilities within the model and diffusion of a physically discrete interface. [15]

1.4 Solid Boundary Models

To include geometric effects, the Baer-Nunziato ADER WENO solver also requires an additional fluid-solid boundary model to simulate the moving geometry. Here, the most common methods for dealing with moving boundaries are identified and used to justify the choice of the modified ghost fluid method within this body of work. There are two main classifications of fluid-solid interaction methods: monolithic and partitioned approaches.

Monolithic vs Partitioned Approaches

Monolithic approaches solve the fluid-solid equations in parallel, by deriving a combined equation that describes both the solid and fluid phases. This can produce highly accurate results, especially where fluid flow affects the movement of the solid as the location of the boundary is dealt with implicitly within the solver. The unified monolithic equations are, however, highly complicated and often require a specialized numerical solver. This is normally only necessary when examining fluid-structure interaction problems, where the flow of the fluid deforms the structure.

The alternative to this is to use a partitioned approach. This involves solving the fluid and solid equations separately with appropriate boundary conditions across the fluid/solid interface, such as a match between pressure, velocity and entropy across the interface.

Using a partitioned approach allows standard, well proven numerical methods to be used for the separate components of the flow. This potentially reduces numerical validation time, increases confidence in results and reduces algorithm development time. The main issue with a partitioned approach is accurately tracking the fluid/solid interface, which is not implicitly tracked as in a monolithic approach. Instead, a separate algorithm, such as the modified ghost fluid method, must be used to track the location of the interface, which can often introduce large errors into the system. [22][23]

The Modified Ghost Fluid Method

One of the most challenging aspects of flow simulation in complex geometries is accurately tracking the fluid/solid interface and there are two main approaches that can be taken, Lagrangian and Eulerian. The Lagrangian approach casts the governing equations in the frame of reference of the moving fluid, meaning that as the fluid flows, so does the computational mesh. This moving mesh implicitly tracks the location of the solid-fluid interface as the boundary will always be adjacent to a specific mesh element. Whilst geometrically accurate, large scale solid movement or high vorticity flows will significantly distort the mesh, thereby reducing the resolution of the simulation. Alternatively, the Eulerian approach uses a fixed (not necessarily ordered) mesh, retaining a high order of accuracy for each individual elements, but significant geometric distortion, particularly for moving meshes.

The original 'Ghost Fluid Method' (GFM) was proposed by Fedkiw et al.[24] as a way to combine the qualities of both the Eulerian and Lagrangian approaches through use of a level set function. Its modification allows an interpolation function to reflect values of pressure, density and velocity (for both phases) at the zero level-set, the fluid-solid interface. The main advantage of the MGFM means that the underlying solver does not need to be modified, instead the boundaries can be applied as a separate module within the code.

The Combined Solver

Following the above review, it was decided to user a Baer-Nunziato multiphase flow model, combined with the MGFM algorithm for integrating moving solid boundaries. There were several reasons behind this choice. (i) A standard solver can be used for the Baer-Nunziato equations, (ii) it is fairly easy to implement any arbitrary geometry, (iii) the two solvers can be validated independently before being validated together, expediting algorithm development time. The following section examines the mathematical properties of the Baer-Nunziato equations and how these can be incorporated into a numerical solution algorithm.

1.5 Compressible Hyperbolic Systems

The Baer-Nunziato equations form a hyperbolic, non-linear system of PDE's and as such, exhibit wave-like characteristics in their solution. Specifically this means that speed at which information propagates through the domain is finite and thus a disturbance in the initial data will not instantaneously act at other points within the system. This has several ramifications for both the solution method and the way it can be implemented.

The Finite Volume Method

Hyperbolic systems generally require more sophisticated solution methods than their Parabolic or Elliptic counterparts. This is due to the existence of so-called 'weak solutions', better known as shock waves and exhibited as sharp discontinuities within the vector field. The high gradients exhibited by these weak solutions mean that standard finite difference methods incorrectly predict wave propagation speed and will induce significant diffusion of sharp discontinuities. This can lead to an incorrect solution. Instead an integral formulation, such the finite volume method, must be used, exact details of this formulation can be found in section 2.

There is significant literature on high order numerical solutions of the finite volume scheme, many of which are reviewed in Toro [25]. However as mentioned previously, use of the Baer-Nunziato equations often leads to incorrect diffusion of the interface between the two phases. In order to counter this, an arbitrarily high order method is used - the ADER-WENO scheme, modified for non-conservative systems.

The finite speed of information propogation also means that hyperbolic systems are amenable to parallelization.

Parallelization

Within the last 15 years, single core processor speeds have begun to significantly plateau and as a counter-measure to this, there has been significant growth in the areas of multi-core and general purpose GPU processing. Whilst this means that there is often an order of magnitude of computing power available to multicore CPU's over single core units, it is difficult to utilize due to the latency in exchanging information between individual cores. Parallelizing Elliptic and Parabolic PDE solvers proves very difficult because of this latency. These PDE's require information exchange between all points in the computational grid (often in the form of an iterative update formula) and may therefore require multiple large information exchanges every time step. Hyperbolic equations on the other hand do not suffer from this. The finite speed of information propagation means that at each time step, each computational cell may only be required to communicate with a handful of its neighboring cells (dependent on the order of the discretization). This means low levels of information exchange between cores and thus lower levels of latency and greater speedup.

1.6 Review of the ADER-WENO method

One of the pitfalls of the BN model is that it requires a high resolution, shock-capturing numerical method, such as an ADER-WENO scheme used in this work. In short, the ADER-WENO method uses a non-linear weighting function to generate a space-time shock-capturing polynomial approximation to the BN PDE that minimizes numerical dispersion. This polynomial can be integrated over space and time to produce an update to the finite volume numerical scheme.

The origins of the ADER-WENO scheme can be traced back to van Leer and Harten et al. [26], who developed the use of fully discrete one-step time update schemes over the original, more complicated multi-stage integrator schemes. In their schemes, a low order polynomial would be reconstructed using the piecewise data from the finite volume update scheme and evaluated at element interfaces to generate and solve an approximate Riemann problem, e.g. in the MUSCL and ENO scheme. This was also the basis of the idea behind the original ADER scheme, whereby a polynomial would be reconstructed over each element at time t^n and then used in the Cauchy-Kovalewski procedure to approximate temporal derivatives by Taylor expansion. However, for complicated non-linear systems, such as the BN equations, the Cauchy-Kowalewski procedure was impractical and so was replaced by an element-local space-time discontinuous Galerkin predictor, based on the weak form of the underlying PDE.

The ADER-WENO is effectively a combination of two underlying methods, the WENO scheme, which produces element wise reconstructed polynomials of arbitrary order, such that they minimize the effects of Runge's phenomenon. Before using this reconstruction within a discontinuous Galerkin (DG) predictor to generate a polynomial over both space and time for each element. This space-time polynomial can then be used to generate and solve a Riemann problem at the edges of each cell, using any arbitrary Riemann solver. One advantage of the ADER-WENO system is that it is applicable to any conservative/non-conservative system of hyperbolic equations, so once the initial framework has been written, it can be easily adapted to e.g. the Euler equations, the BN equations or even the shallow water wave equations. For a more in depth discussion and review of the ADER-WENO scheme, see Dumbser et al. [4]

Sections 4.1 and 4.2 demonstrate the high order of accuracy of the ADER-WENO method for both conservative and non-conservative systems of equations, which is particularly when paired with a high accuracy Riemann solver, such as the Osher-Solomon method [27], described in Section 3.4.

1.7 Summary

The conclusion of the literature review was to use an ADER-WENO method to numerically solve the Baer-Nunziato equations of multiphase flow. This would be used in conjunction with a modified ghost fluid method to apply solid moving boundary conditions. Section 2 discusses the Baer-Nunziato and ghost fluid models in further depth, before the method used for their numerical solution is described in Section 3.

2 Mathematical Models

Within this section, the mathematical models selected within the literature review are discussed, along with the additional models used to describe the various equations of state and phase interactions required to form a complete description of a multiphase granular flow. Unless otherwise stated all terms are given as dimensionless quantities.

2.1 The Primitive Baer-Nunziato Model

The Baer-Nunziato Model was first proposed by Baer and Nunziato [15] as a method for modeling granular detonation mechanics. It comes as a direct consequence of consideration of conservation of mass, momentum, energy and total phase quantity and can therefore be applied to any generalized multiphase flow, including granular ones. The two dimensional, two phase BN equations are a system of 9 hyperbolic, non-conservative, non-linear partial differential equations (PDE's) and can most succinctly be written in partial conservation variable form as:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) + \mathbf{B}(\mathbf{U})\nabla \cdot \mathbf{U} = \mathbf{S}(\mathbf{U}) \tag{1}$$

Where $\mathbf{U} = \{\alpha_1, \alpha_1\rho_1, \alpha_1\rho_1 u_1, \alpha_1\rho_1 v_1 \alpha_1\rho_1 E_1, \alpha_2\rho_2, \alpha_2\rho_2 u_2, \alpha_2\rho_2 v_2, \alpha_2\rho_2 E_2\}$. The conservative part $\nabla \cdot \mathbf{F}(\mathbf{U})$ contains the terms for conservation of mass, momentum and energy, the non-conservative hyperbolic part, $\mathbf{B}(\mathbf{U})\nabla \cdot \mathbf{U}$ contains the terms describing advection and pressure differentials of the two-phase interface. The source terms $\mathbf{S}(\mathbf{U})$ describe the drag, pressure relaxation and compaction terms between the two phases as well as any additional stiff reaction terms or viscosity terms. The non source terms are given by:

$$\mathbf{F}_x(\mathbf{U}) = \begin{bmatrix} \alpha_1 \\ \alpha_1\rho_1 u_1 \\ \alpha_1\rho_1(u_1^2 + p_1) \\ \alpha_1\rho_1 u_1 v_1 \\ \alpha_1\rho_1 E_1 \\ \alpha_2\rho_2 u_2 \\ \alpha_2\rho_2(u_2^2 + p_2) \\ \alpha_2\rho_2 u_2 v_2 \\ \alpha_2\rho_2 E_2 \end{bmatrix} \qquad \mathbf{F}_y(\mathbf{U}) = \begin{bmatrix} \alpha_1 \\ \alpha_1\rho_1 v_1 \\ \alpha_1\rho_1 u_1 v_1 \\ \alpha_1\rho_1(v_1^2 + p_1) \\ \alpha_1\rho_1 E_1 \\ \alpha_2\rho_2 v_2 \\ \alpha_2\rho_2 u_2 v_2 \\ \alpha_2\rho_2(v_2^2 + p_2) \\ \alpha_2\rho_2 E_2 \end{bmatrix} \tag{2}$$

and non-conservative vectors given by:

$$\mathbf{B}_x(\mathbf{U}) = \begin{bmatrix} u_1 \\ 0 \\ -p_2 \\ 0 \\ -p_2u_1 \\ 0 \\ p_2 \\ 0 \\ p_2u_1 \end{bmatrix} \qquad \mathbf{B}_y(\mathbf{U}) = \begin{bmatrix} v_1 \\ 0 \\ 0 \\ -p_2 \\ -p_2v_1 \\ 0 \\ 0 \\ p_2 \\ p_2u_1 \end{bmatrix} \tag{3}$$

The source terms, $\mathbf{S}(x, y, t)$, are strongly dependent on phase properties and as such are unique to a given phase pairing, various models of these are discussed in Section 2.2. The BN equations can also be written in a primitive variable form due to algorithmic considerations. Firstly, fewer terms are required to be generated in the DG solver, increasing speed and secondly, a more efficient generation of the Eigendecomposition can be done. This important for the Osher-Solomon Riemann solver described in Section 3.4. The primitive BN equations are written as,

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{F}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial x} + \mathbf{G}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial y} = \mathbf{S}(x, y, t) \tag{4}$$

Where $\mathbf{w}^T = [\alpha_1, \rho_1, u_1, v_1, p_1, \rho_2, u_2, v_2, p_2]$. $\alpha_1 \in [0, 1]$ is the volume fraction of phase 1, ρ is density, u is horizontal velocity, v is vertical velocity and p is the phase pressure. The subscripts 1 and 2 refer to the phase that the variable belongs to.

The matrices $\mathbf{F}(\mathbf{w})$ and $\mathbf{G}(\mathbf{w})$ are given by:

$$\mathbf{F}(\mathbf{w}) = \begin{bmatrix} u_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & u_1 & \rho_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\Delta p/(\alpha_1\rho_1) & 0 & u_1 & 0 & 1/\rho_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_1^2\rho_1 & 0 & u_1 & 0 & 0 & 0 & 0 \\ (\Delta u\rho_2)/(\alpha_1 - 1) & 0 & 0 & 0 & 0 & u_2 & \rho_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u_2 & 0 & \rho_2^{-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_2 & 0 \\ (a_2^2\Delta u\rho_2)/(\alpha_1 - 1) & 0 & 0 & 0 & 0 & 0 & a_2^2\rho_2 & 0 & u_2 \end{bmatrix} \tag{5}$$

$$\mathbf{G}(\mathbf{w}) = \begin{bmatrix} v_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_1 & 0 & \rho_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\Delta p/(\alpha_1\rho_1) & 0 & 0 & v_1 & \rho_1^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1^2\rho_1 & v_1 & 0 & 0 & 0 & 0 \\ (\Delta v\rho_2)/(\alpha_1 - 1) & 0 & 0 & 0 & 0 & v_2 & 0 & \rho_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_2 & \rho_2^{-1} \\ (a2^2\Delta v\rho_2)/(\alpha_1 - 1) & 0 & 0 & 0 & 0 & 0 & 0 & a_2^2\rho_2 & v_2 \end{bmatrix} \tag{6}$$

Here $\Delta u = u_2 - u_1$, $\Delta v = v_2 - v_1$, $\Delta p = p_1 - p_2$ and a_k refers to the speed of sound in phases 1 and 2. [28]

The Baer-Nunziato equations do not form a complete description of the multiphase interaction - interface velocity and pressure also need to be defined in terms of the two phase properties. The model proposed by Dumbser et al. [29] was used within this work. The interface velocity is set to be equal to the velocity of the first phase at the interface (the stiffer phase) and inter-phase pressure is equal to the secondary phase's pressure. Other fully symmetric interface models are available, but are often mathematically ill-posed [2].

Eigendecomposition

It is often convenient to write the Baer-Nunziato equations in eigendecomposition form in order to generate approximate Riemann solvers through linearisations (eg. Roe HLLC-type [28]) or approximation in phase space.

$$\frac{\partial \mathbf{w}}{\partial t} + \Phi_{\mathcal{F}}(\mathbf{w})\Lambda_{\mathcal{F}}(\mathbf{w})\Phi_{\mathcal{F}}^{-1}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial x} + \Phi_{\mathcal{G}}(\mathbf{w})\Lambda_{\mathcal{G}}(\mathbf{w})\Phi_{\mathcal{G}}^{-1}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial y} = \mathbf{S}(x, y, t) \tag{7}$$

with a standard eigendecomposition of the form:

$$\mathbf{F}(\mathbf{w}) = \Phi_{\mathcal{F}}(\mathbf{w})\Lambda_{\mathcal{F}}(\mathbf{w})\Phi_{\mathcal{F}}^{-1}(\mathbf{w}) \quad , \quad \mathbf{G}(\mathbf{w}) = \Phi_{\mathcal{G}}(\mathbf{w})\Lambda_{\mathcal{G}}(\mathbf{w})\Phi_{\mathcal{G}}^{-1}(\mathbf{w}) \tag{8}$$

These matrices of eigenvectors and eigenvalues have a close form solution, given by:

$$\Phi_{\mathcal{F}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho_1 & 1 & 0 & \rho_1 & 0 & 0 & 0 & 0 \\ 0 & -a_1 & 0 & 0 & a_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \Delta p/\alpha_1 & a_1^2\rho_1 & 0 & 0 & a_1^2\rho_1 & 0 & 0 & 0 & 0 \\ -(\Delta u^2\rho_2)/((a_2^2 - \Delta u^2)(\alpha_2)) & 0 & 0 & 0 & 0 & \rho_2 & 1 & 0 & \rho_2 \\ (a_2^2\Delta u)/((a_2^2 - \Delta u^2)(\alpha_2)) & 0 & 0 & 0 & 0 & -a_2 & 0 & 0 & a_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -(a_2^2\Delta u^2\rho_2)/((a_2^2 - \Delta u^2)(\alpha_2)) & 0 & 0 & 0 & 0 & a_2^2\rho_2 & 0 & 0 & a_2^2\rho_2 \end{bmatrix} \tag{9}$$

$\Phi_{\mathcal{F}}$ has corresponding diagonal eigenvalue matrix: $\Lambda_{\mathcal{F},ii} = \{u_1, u_1 - a_1, u_1, u_1, u_1 + a_1, u_2 - a_2, u_2, u_2, u_2 + a_2\}^T$

$$\Phi_{\mathcal{G}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho_1 & 0 & 1 & \rho_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_1 & 0 & 0 & a_1 & 0 & 0 & 0 & 0 \\ \Delta p/\alpha_1 & a_1^2\rho_1 & 0 & 0 & a_1\rho_1 & 0 & 0 & 0 & 0 \\ -(\Delta v^2\rho_2)/((a_2^2 - \Delta v^2)(\alpha_2)) & 0 & 0 & 0 & 0 & \rho_2 & 0 & 1 & \rho_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ (a_2^2 dv)/((a2^2 - \Delta v^2)(\alpha_2)) & 0 & 0 & 0 & 0 & -a_2 & 0 & 0 & a_2 \\ -(a_2^2\Delta v^2\rho_2)/((a_2^2 - \Delta v^2)(\alpha_2)) & 0 & 0 & 0 & 0 & a_2^2\rho_2 & 0 & 0 & a_2^2\rho_2 \end{bmatrix} \qquad (10)$$

$\Phi_{\mathcal{G}}$ has corresponding diagonal eigenvalue matrix: $\Lambda_{\mathcal{G},ii} = \{v_1, v_1 - a_1, v_1, v_1, v_1 + a_1, v_2 - a_2, v_2, v_2, v_2 + a_2\}^T$ [28]

The following subsections described the additional closure models: source exchange terms and equations of state which can be used to described various multiphase granular systems.

2.2 Source Exchange terms

The source exchange terms describe the exchange of density, velocity and volume fraction between phases and can therefore be used to induce drag, heat exchange, porosity change and surface tension between the two phases. Derivation of these exchange terms is most simply done within the conservative variable formulation, meaning that in order to be used in the primitive formulation, they must be multiplied by the inverse of the Jacobian matrix given by:

$$J_{ij}(\mathbf{w}) = \frac{\partial U_i}{\partial w_j} \qquad (11)$$

The complicated and varied nature of phase interactions mean that the corresponding exchange terms are highly complex. Instead of using large, generic models, the source terms are often case dependent and come from a dimensional or asymptotic analysis of the problem at hand. In this work, three variations of the exchange model are used, coming from various assumptions of the strength of the compaction wave. Here, exchange models are listed from the stronger compaction wave models to the weakest, with their assumptions discussed. Again these are given in dimensionless form, usually with some additional scaling required.

Schwendeman's Strong Compaction Wave Model

The model of Schwendeman et al. [5] describes compaction within a high porosity, high Mach number gaseous/granular flow in their case for a $\alpha_{1,0} = 0.73$, $Ma = 100$ flow. The model comes from the assumption that the compaction is driven mainly through strong variation in the packing potential energy - $B(\alpha_1)$ and the exchange of thermal energy between phases to achieve equilibrium. Additionally, drag is assumed to be linearly proportional to the difference in velocities between phases. The exchange process is described through the vector of source terms:

$$\mathbf{S}(\mathbf{w}) = \mathbf{J}^{-1}(\mathbf{w}) \quad [\mathcal{C}, 0, \mathcal{M}_u, \mathcal{M}_v, \mathcal{E} - p_2\mathcal{C}, 0, -\mathcal{M}_u, -\mathcal{M}_v, p_2\mathcal{C} - \mathcal{E}]^T \tag{12}$$

with relaxation parameters:

$$\mathcal{C} = \frac{1}{\mu_c}\big[\alpha_s(1-\alpha_s)(p_s - p_g - \beta)\big] \tag{13}$$

$$\beta(\alpha_s, \rho_s) = \alpha_s \rho_s \frac{dB(\alpha_s)}{d\alpha_s} \tag{14}$$

Where $B(\alpha_s)$ is the compaction potential energy and $\mathcal{M}_u = \delta(u_g - u_s)$, $\mathcal{M}_v = \delta(v_g - v_s)$, $\mathcal{E} = -\mathcal{M}_u u_s - \mathcal{M}_v v_s$. The compaction tensor, β is given by:

$$\beta = -(p_{2,0} - p_{1,0})\frac{\alpha_1\rho_1}{\alpha_{1,0}\rho_{1,0}}\Big(\frac{2-\alpha_{1,0}}{2-\alpha_1}\Big)\frac{ln(1-\alpha_1)}{ln(1-\alpha_{1,0})} \tag{15}$$

The drag coefficient, δ is an experimentally-fit parameter, often taken to be 2^2.

Rogue's Weak Compaction Model

This model described in Rogue et al. [30] describes a weak compaction wave produced by granular dispersion for a low volume fraction, Low Mach number gas/granular flow. They assume incompressibility for the granular phase and a linear packing potential energy, linearized around the background flow such that $\alpha_1 p_2 \approx \alpha_{1,0} p_{2,0}$ and there is therefore a negligible gas pressure variation.

$$\mathbf{S}(\mathbf{w}) = \mathbf{J}^{-1}(\mathbf{w})\{0, 0, F_u, F_v, u_1F_u + v_1F_v + Q, 0, -F_u, -F_v, -u_1F_u - v_1F_v - Q\} \tag{16}$$

with drag forces in the horizontal and vertical directions given by:

$$F_u = \frac{3}{4}C_d\frac{\rho_2}{d_p}\alpha_1|u_2 - u_1|(u_2 - u_1) \tag{17}$$

$$F_v = \frac{3}{4}C_d\frac{\rho_2}{d_p}\alpha_1|v_2 - v_1|(v_2 - v_1) \tag{18}$$

and thermal energy exchange:

$$Q = \pi d_p \alpha_1 K Re^{0.7} Pr^{0.33} (T_2 - T_1) \tag{19}$$

Where Re is the relative Reynolds number and Pr is the Prandtl number. Rogue et al. assumed that the solid was incompressible and derived an equation describing the advection of the solid by combining the homogeneous parts of the momentum and mass conservation equations (in 1D).

$$\frac{\partial u_s}{\partial t} + u_s \frac{\partial u_s}{\partial x} + \frac{a_s^2}{\alpha_s \rho_s} \frac{\partial \alpha_s \rho_s}{\partial x} \tag{20}$$

With the solid speed of sound given by:

$$a_s^2 = \frac{\alpha_{1,0} p_{2,0}}{\rho_1} + a_{s,0}^2 \tag{21}$$

Where $\alpha_{1,0}$ is the upstream solid volume fraction, $p_{2,0}$ is the upstream gas pressure and $a_{s,0}$ is the upstream speed of sound in the solid. In terms of the Schwendeman model, this is equivalent to having a compaction energy term of:

$$\beta = \rho_1 a_{s,0}^2 (1 - \frac{\alpha_{1,0}}{\alpha_1}) \tag{22}$$

Within this work, the incompressibility constraint was ignored and instead a stiffened gas equation of state was used to model solid phases.

Saurel's pressure equilibrium model

This model is taken from Saurel [2] and describes the forces of surface tension and drag between phases. It provides the most general framework of the models described here for modeling pressure, velocity and energy exchange between phases. By altering the relaxation viscosity μ_r, the effects of surface tension, granular compaction, and equilibrium multiphase flows can be modeled. Using an assumption of a very weak compaction wave in a compressible/compressible gas/granular flow yields a non-varying packing potential energy. This gives a complete pressure equilibrium of $p_1 = p_2$. To implement this the relaxation viscosity is set as the limit $\mu_r \to \infty$ and thus the source terms are run to a steady state.

$$\begin{aligned}\mathbf{S}(\mathbf{w}) = \mathbf{J}(\mathbf{w})^{-1}[\mu_r(p_1 - p_2), 0, F_u, F_v, \mu_r p_1(p_1 - p_2) + uF_u + vF_v, 0, -F_u,\\ - F_v, \mu_r p_1(p_2 - p_1) - u_2 F_u - v_2 F_v]^T\end{aligned} \tag{23}$$

Here the same granular drag parameter is used as Rogue's model.

2.3 Equations of state

Both the Euler equations and the Baer-Nunziato equations require additional closure laws to model the relationship between internal energy, pressure and density for each phase, otherwise known as an equation of state. There are a myriad of potential options, however here just two linear equations of state are examined, that provide an accurate approximation for various forms of inert, low Mach number two phase flow.

The stiffened gas equation

The stiffened gas equation of state is a simplification of the Mie–Gruneisen equation of state, originally used to describe the evolution of pressure in a shock compressed solid. Due to its three empirical parameters, it can be used as a linear approximation to any arbitrary equation of state and thus can roughly model liquids, gases and solids, provided there is only limited variation in pressure and density. The stiffened gas EOS is given by:

$$e(\rho, p) = \frac{p + \gamma\bar{\pi}}{(\gamma - 1)\rho} + q \tag{24}$$

Where γ is the ratio of specific heats, $\bar{\pi}$ is a stiffening pressure and q is a heat release parameter. Menikoff [31] contains a useful list of empirically derived parameters for modeling various common materials, such as water, air and various metals. The speed of sound, c through a stiffened gas is given by: [28]

$$c^2 = \frac{\gamma(p + \bar{\pi})}{\rho} \tag{25}$$

The Virial Gas equation

The Virial equation of state comes from a first order expansion of the isotropic gas law and thus can provide a linear approximation for any isotropic equation of state. The EOS is given by: [2]

$$e(\rho, p) = \frac{p}{(\gamma - 1)(1 + b\rho)\rho} \tag{26}$$

γ is the ratio of specific heats of the gas and b is a data-fit parameter, equal to 1.0×10^{-3} in air. The speed of sound within a Virial gas is given by:

$$c^2 = \frac{\gamma p}{\rho}\big[1 + b\rho + \frac{b^2\rho^2}{\gamma(1 + b\rho)}\big] \tag{27}$$

The previous sections give a full description of a multiphase granular flow within an unbounded space. However, as described within the introduction, one of the main aims of the project was to develop a multiphase model that could be used within moving boundaries. The following section describes the two boundary conditions used within this work.

2.4 Boundary Conditions

Solid Boundaries

For High-Reynolds number flows, the boundary layer,δ induced within the fluid near the walls will be small as $\delta \approx Re^{-\frac{1}{2}}$ [32] and thus can be negated. This effectively gives a slip/reflective boundary condition at solid walls, equivalent to a reflective boundary condition for velocity and a 'pass through' condition for other variables. For a phase velocity vector, $\mathbf{u}_k$, this can be written as:

$$\hat{\mathbf{n}} \cdot \nabla \mathbf{u}_k \big(s(x, y)\big) = 0 \tag{28}$$

Whilst the condition for non-vector quantities: phase pressure, phase density and volume fraction is given by:

$$\frac{\partial J}{\partial \hat{\mathbf{n}}} \big(s(x, y)\big) = 0 \tag{29}$$

Where in both cases $\hat{\mathbf{n}}$ is the normal vector to the boundary wall and s(x,y) is the parametric function describing the geometry of the solid boundary. These set of boundary equations are used to both describe fixed solid walls and the process happening within the ghost-fluid boundary, the numerical discretization of which can be found in Section 3.7.

Pass Through Conditions

The pass through conditions refer to a Neumann boundary condition where the derivative is set to 0. In this work, this is used to approximate the boundaries of a domain which are open to outside the system, allowing transport of mass, momentum and energy to outside of the enclosed system. In this, velocity is not reflected at the boundary, instead it is allowed to 'pass through' and thus every variable is defined by equation (29) at the boundary.

2.5 Summary

The models described within this section give a complete mathematical approximation to the types of multiphase granular flows that this work aims to solve. The following section describes in detail the high order ADER-WENO method used to solve the Baer-Nunziato equations, as well as the modified ghost fluid method used to apply boundary conditions.

3 Methods

The non-linear, non-conservative nature of the Baer-Nunziato equations provide a challenging problem in developing accurate numerical methods. The shock waves present in the solution cannot accurately be resolved with a standard finite difference scheme, whilst the non-conservative terms mean that many standard approximate Riemann solver methods cannot be used without significant modification. Solutions to the BN equations also often become unstable if the numerical method used produces significant numerical diffusion. In this section, the complete ADER-WENO/MGFM algorithm is outlined and discussed.

3.1 The finite volume Baer Nunziato Formulation

The finite volume formulation of the primitive variable, two phase, two dimensional Baer-Nunziato equations can be derived from the original PDE formulation found in (2.1):

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{F}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial x} + \mathbf{G}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial y} = \mathbf{S}(x, y, t) \tag{30}$$

Where $\mathbf{w}^T = [\alpha_1, \rho_1, u_1, v_1, p_1, \rho_2, u_2, v_2, p_2]$, with $\alpha_1 \in [0, 1]$ being the volume fraction of phase 1.

Integrating over a volume in space and time, $I_{ij}^n = [t^{n+1}; t^n] \times [x_{i-\frac{1}{2}}; x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}; y_{j+\frac{1}{2}}]$ yields the finite volume discretization:

$$\bar{\mathbf{w}}^{n+1} = \bar{\mathbf{w}}^n + \Delta t(\bar{\mathbf{S}}_{i,j} - \bar{\mathbf{F}}_{i,j} - \bar{\mathbf{G}}_{i,j}) \tag{31}$$

where

$$\bar{\mathbf{w}}_{i,j}^n = \frac{1}{\Delta y \Delta x} \int_{x_{i-\frac{1}{2},j}}^{x_{i+\frac{1}{2},j}} \int_{y_{i,j-\frac{1}{2}}}^{y_{i,j+\frac{1}{2}}} \mathbf{w}(x, y, t) dx dy \tag{32}$$

$$\bar{\mathbf{F}}_{i,j} = \frac{1}{\Delta t \Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{x_{i-\frac{1}{2},j}}^{x_{i+\frac{1}{2},j}} \int_{y_{i,j-\frac{1}{2}}}^{y_{i,j+\frac{1}{2}}} \mathbf{F}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x} dt dx dy \tag{33}$$

$$\bar{\mathbf{G}}_{i,j} = \frac{1}{\Delta t \Delta x \Delta y} \int_{t^n}^{t^{n+1}} \int_{x_{i-\frac{1}{2},j}}^{x_{i+\frac{1}{2},j}} \int_{y_{i,j-\frac{1}{2}}}^{y_{i,j+\frac{1}{2}}} \mathbf{G}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial y} dt dx dy \tag{34}$$

Evaluation of the non-conservative terms (33) and (34) can be done through separating each integral into a contribution from within the the finite volume and a contribution from the exterior of the finite volume (i.e. the flux when there is a discontinuity present). The integration of the non-smooth parts is defined as a Borel measure through use of Dal Maso-Le Floch-Murat theory, this gives an overall finite volume discretisation as:

$$\bar{\mathbf{w}}^{n+1} = \bar{\mathbf{w}}^{n} - \frac{\Delta t}{\Delta x}[\mathbf{P}^{-}_{i+\frac{1}{2},j} + \mathbf{P}^{+}_{i-\frac{1}{2},j}] - \frac{\Delta t}{\Delta y}[\mathbf{Q}^{-}_{i,j+\frac{1}{2}} + \mathbf{Q}^{+}_{i,j-\frac{1}{2}}] + \Delta t(\bar{\mathbf{S}}_{i,j} - \bar{\mathbf{F}}_{i,j} - \bar{\mathbf{G}}_{i,j}) \quad (35)$$

with the condition:

$$\mathbf{P}^{+}_{i+\frac{1}{2},j} + \mathbf{P}^{-}_{i+\frac{1}{2},j} = \frac{1}{\Delta t \Delta y}\int_{t^n}^{t^{n+1}}\int_{y_{i,j-\frac{1}{2}}}^{y_{i,j+\frac{1}{2}}} \mathcal{P}(\mathbf{w}^{+}(x_{i+\frac{1}{2}}, y, t), \mathbf{w}^{-}(x_{i+\frac{1}{2}}, y, t))dtdy \quad (36)$$

$$\mathbf{Q}^{+}_{i,j+\frac{1}{2}} + \mathbf{Q}^{-}_{i,j+\frac{1}{2}} = \frac{1}{\Delta t \Delta x}\int_{t^n}^{t^{n+1}}\int_{x_{i-\frac{1}{2},j}}^{x_{i+\frac{1}{2},j}} \mathcal{Q}(\mathbf{w}^{+}(x, y_{j+\frac{1}{2}}, t), \mathbf{w}^{-}(x, y_{j+\frac{1}{2}}, t))dtdx \quad (37)$$

$\mathcal{P}$ and $\mathcal{Q}$ are defined as the integral of the non-conservative products (33) and (34) through the phase space connecting the left and right hand sates. For a completely smooth solution, there will be a null phase space connecting the two sates (as they are identical) and thus the finite volume equation will revert to (32). The integrals through phase space are defined as:

$$\mathcal{P}(\mathbf{w}^{+}(x_{i+\frac{1}{2}}, y, t), \mathbf{w}^{-}(x_{i+\frac{1}{2}}, y, t)) = \int_0^1 \mathbf{F}(\Psi(\mathbf{w}^{+}, \mathbf{w}^{-}, s))\frac{\partial}{\partial s}\Psi(\mathbf{w}^{+}, \mathbf{w}^{-}, s)ds \quad (38)$$

$$\mathcal{Q}(\mathbf{w}^{+}(x, y_{j+\frac{1}{2}}, t), \mathbf{w}^{-}(x, y_{j+\frac{1}{2}}, t)) = \int_0^1 \mathbf{G}(\Psi(\mathbf{w}^{+}, \mathbf{w}^{-}, s))\frac{\partial}{\partial s}\Psi(\mathbf{w}^{+}, \mathbf{w}^{-}, s)ds \quad (39)$$

Finally, using the matrix absolute value, defined through the Eigendecompostion of $\mathbf{F}$ and $\mathbf{G}$, $\mathbf{P}^{+}$ and $\mathbf{P}^{-}$ can be separated into positive and negative contributions to feedback into (35). The absolute value of a matrix is defined as:

$$|\mathbf{A}| = \Phi|\Lambda|\Phi^{-1} \quad (40)$$

Where Φ is a column-wise matrix of the eigenvectors of A, Φ^{-1} is its inverse and $|\Lambda|$ is a diagonal matrix containing the absolute value of each eigenvalue corresponding to each column eigenvector in Φ. Thus the positive and negative states of $\mathbf{P}$ and $\mathbf{Q}$ can be found through:

$$\mathbf{P}^{\pm}_{i+\frac{1}{2},j} = \frac{1}{\Delta t \Delta y}\int_{t^n}^{t^{n+1}}\int_{y_{i,j-\frac{1}{2}}}^{y_{i,j+\frac{1}{2}}} \mathcal{P}^{\pm}(\mathbf{w}^{+}(x_{i+\frac{1}{2}}, y, t), \mathbf{w}^{-}(x_{i+\frac{1}{2}}, y, t))dtdy \quad (41)$$

$$\mathcal{P}^{\pm}(\mathbf{w}^{+}(x_{i+\frac{1}{2}}, y, t), \mathbf{w}^{-}(x_{i+\frac{1}{2}}, y, t)) = \\ \frac{1}{2}\int_0^1 \Big(|\mathbf{F}(\Psi(\mathbf{w}^{+}, \mathbf{w}^{-}, s))| \mp \mathbf{F}(\Psi(\mathbf{w}^{+}, \mathbf{w}^{-}, s))\Big)\frac{\partial}{\partial s}\Psi(\mathbf{w}^{+}, \mathbf{w}^{-}, s)ds \quad (42)$$

Once a path through phase space has been chosen, equation (14) can be used to generate a whole host of Riemann problem solvers for the primitive Baer-Nunziato equations, e.g. through a Roe linearization to yield a Roe-type solver [3], or a wave-based solver, akin to the HLLC method for single phase problems. Although fast, these Riemann solvers require linearisation around a fixed state (often $M_a = 1$) and will therefore introduce large errors for Low-Mach number problems. For this paper, a fully non-linear type solver will therefore be used - the Osher-Solomon solver. Here the phase-space path is set as the most basic $s = \mathbf{w}^- + s(\mathbf{w}^+ - \mathbf{w}^-)$ and equation (14) is evaluated using a Gauss-Legendre quadrature. Although this is significantly more accurate in the Low-Mach number regime, this adds significant computational overhead as each evaluation of $\mathcal{P}$ and $\mathcal{Q}$ will require the calculation and inversion of the set of eigenvectors/values of $\mathbf{F}$ and $\mathbf{G}$, N times, where N is the number of Gaussian points of the scheme. However, as discussed in Section 1.5, this type of scheme is amenable to parallelization.

3.2 The WENO reconstruction

The WENO (weighted-essentially-non-oscillatory) method is used to construct Nth order degree polynomial approximations from averaged variables (i.e. those found in the finite volume update formula). It constructs several polynomial approximations across each cell and non-linearly combines them to produce high order polynomials in smooth regions and low order, non-oscillatory polynomials when shock waves are present. The reconstruction process can be seen in Figure 3, where non-oscillating polynomials are generated for each adjacent cell.

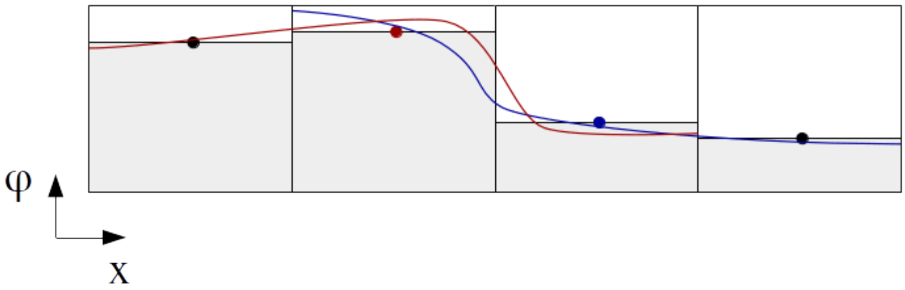


Figure 3: Diagram of the WENO reconstruction for two piecewise data points. Cell 2 (red) and cell 3 (blue) use the cells within their own local neighborhood to each generate an interpolating polynomial (the curved lines) for a variable ϕ. The two polynomials predict different values of ϕ at the interface between the two cells, thereby generating a Riemann problem.

Initially, each computational cell is rescaled such that $\zeta = (x - x_{i-\frac{1}{2}})/\Delta x$, $\eta = (y - y_{j-\frac{1}{2}})/\Delta y$ and a basis over this new coordinate space is defined, consisting of N+1 Lagrange polynomials, ψ_l, acting over the interval $[0, 1]$, with the property: $\psi_l(\zeta_k) = \delta_{lk}$. The weighting attached to each of these polynomials can be found through integration of each polynomial over a set of stencils, $\mathcal{S}_{ij}^{x,s}$:

$$\mathcal{S}_{ij}^{x,s} = \bigcup_{e=i-L(s)}^{i+R(s)} I_{ej} \qquad \mathcal{S}_{ij}^{y,s} = \bigcup_{e=j-L(s)}^{j+R(s)} I_{ie} \tag{43}$$

For odd schemes, three stencils are chosen such that $L(s) = [N/2, N, 0]$, $R(s) = [0, N, N/2]$ and for even schemes: $L(s) = [(N-1)/2, (N+1)/2, N, 0]$ $R(s) = [(N+1)/2, (N-1)/2, 0, N]$

$$\bar{\mathbf{w}}_{ej}^{n} = \int_{e}^{e+1} \mathbf{w}_{ij}^{s,x}(\zeta, t^n) d\zeta \approx \int_{e}^{e+1} \psi_p(\zeta) \hat{\mathbf{w}}_{ij,p}^{n,s} d\zeta + O(\Delta x^{\alpha}) \qquad \forall I_{ej} \in \mathcal{S}_{ij}^{s,x} \tag{44}$$

and solved as a system of linear equations to find $\hat{\mathbf{w}}_{ij,p}^{n,s}$. The polynomials produced by each stencil can then be non-linearly combined using an 'oscillation index'. Essentially this is a non-linear weighting function which gives high weights for smooth polynomials and low weights for highly oscillatory ones, the indicator is given by:

$$\sigma_s = \hat{\mathbf{w}}_{ij,p}^{n,s} \hat{\mathbf{w}}_{ij,m}^{n,s} \sum_{\beta=1}^{N} \int_0^1 \frac{\partial^{\beta} \psi_p(\zeta)}{\partial \zeta^{\beta}} \frac{\partial^{\beta} \psi_m(\zeta)}{\partial \zeta^{\beta}} d\zeta \tag{45}$$

The x-direction reconstructed polynomial can then be given as:

$$\hat{\mathbf{w}}_{ij,p}^{x} = \sum_{s} \left[\frac{\lambda_s}{(\sigma_s + \epsilon)^r} \right]_{\hat{N}} \hat{\mathbf{w}}_{ij,p}^{n,s} \tag{46}$$

Where $[\ ]_{\hat{N}}$ represents the normalization function, λ_s, ϵ and r are arbitrary parameters that control the level of oscillation damping.

This leads to a complete polynomial reconstruction in the x-direction and the process can then be repeated in the y-direction, using $\hat{\mathbf{w}}_{ij,p}^{x}$ as our starting point.

$$\hat{\mathbf{w}}_{ij,p}^{x} = \int_{e}^{e+1} \mathbf{w}_{ij,p}^{s,y}(\eta, t^n) d\eta \approx \int_{e}^{e+1} \psi_q(\eta) \hat{\mathbf{w}}_{ij,pq}^{n,s} d\zeta + O(\Delta y^{\alpha}) \qquad \forall I_{ie} \in \mathcal{S}_{ij}^{s,y} \tag{47}$$

and combine using the same non-linear oscillation index. This gives us the fully reconstructed polynomial over each element at time $t = t^n$.

$$\hat{\mathbf{w}}_{ij}(\zeta, \eta, t^n) = \hat{\mathbf{w}}_{ij,pq}^{n} \psi_p(\zeta) \psi_q(\eta); \tag{48}$$

3.3 The Discontinuous Galerkin Predictor

Initially, the PDE (30) without source terms is rescaled in terms of the basis defined in 3.1 along with time, such that the PDE becomes:

$$\frac{\partial \mathbf{w}}{\partial \tau} + \mathbf{f}^*(\mathbf{w})\frac{\partial \mathbf{w}}{\partial \zeta} + \mathbf{g}^*(\mathbf{w})\frac{\partial \mathbf{w}}{\partial \eta} = 0 \tag{49}$$

A test function: $\theta_{pqr}(\zeta, \eta, \tau) = \psi_p(\zeta)\psi_q(\zeta)\psi_r(\tau)$ is then multiplied by equation (49) and integrated over the finite volume in time to derive the DG integral equation:

$$\int_0^1\int_0^1 \theta_{pqr}(\zeta,\eta,1)\mathbf{w}(\zeta,\eta,1)d\zeta d\eta - \int_0^1\int_0^1\int_0^1 \theta_{pqr}\left[\mathbf{f}^*(\mathbf{w})\frac{\partial \mathbf{w}}{\partial \zeta} + \mathbf{g}^*(\mathbf{w})\frac{\partial \mathbf{w}}{\partial \eta}\right]d\zeta d\eta d\tau - \int_0^1\int_0^1\int_0^1 \frac{\partial \theta_{pqr}}{\partial \tau}\mathbf{w} d\zeta d\eta d\tau = \int_0^1\int_0^1 \theta_{pqr}(\zeta,\eta,0)\mathbf{w}(\zeta,\eta,t^n)d\zeta d\eta \tag{50}$$

Each unknown can then be approximated with a Lagrange polynomial, e.g $\mathbf{f}^*(\mathbf{w})\frac{\partial \mathbf{w}}{\partial \zeta} = f *_{p'q'r'} \theta_{p'q'r'}$, using a multi-index $\mathfrak{p} = \{p, q, r\}$, $\mathfrak{q} = \{p', q', r'\}$, $\mathfrak{r} = \{\hat{p}, \hat{q}, \hat{r}\}$ for convenience, it can be written:

$$\int_0^1\int_0^1 \theta_{\mathfrak{p}'}(\zeta,\eta,1)\theta_{\mathfrak{q}'}(\zeta,\eta,1)d\zeta d\eta \cdot \hat{\mathbf{w}}_{ij,\mathfrak{q}'} - \int_0^1\int_0^1\int_0^1 \theta_{\mathfrak{p}}\theta_{\mathfrak{q}} d\zeta d\eta d\tau \cdot \left[\mathbf{f}^*(\hat{\mathbf{w}}_{ij,\mathfrak{q}}) + \mathbf{g}^*(\hat{\mathbf{w}}_{ij,\mathfrak{q}})\right] - \int_0^1\int_0^1\int_0^1 \frac{\partial \theta_{\mathfrak{p}}}{\partial \tau}\theta_{\mathfrak{q}} d\zeta d\eta d\tau \cdot \hat{\mathbf{w}}_{ij,\mathfrak{q}} = \int_0^1\int_0^1 \theta_{\mathfrak{p}'}(\zeta,\eta,0)\theta_{\mathfrak{q}'}(\zeta,\eta,0)d\zeta d\eta \cdot \hat{\mathbf{w}}^n_{ij,\mathfrak{q}'} \tag{51}$$

$\mathbf{f}^*(\hat{\mathbf{w}}_{ij,\mathfrak{q}})$ and $\mathbf{g}^*(\hat{\mathbf{w}}_{ij,\mathfrak{q}})$ can be discretised as follows:

$$\mathbf{f}^*(\hat{\mathbf{w}}_{ij,\mathfrak{q}}) = \mathbf{f}^*(\hat{\mathbf{w}}_{ij,\mathfrak{q}})\frac{\partial \theta_{\mathfrak{r}}(\zeta_{\mathfrak{q}}, \eta_{\mathfrak{q}}, \tau_{\mathfrak{q}})}{\partial \zeta}\hat{\mathbf{w}}_{ij,\mathfrak{r}} \tag{52}$$

Which is solved via an iterative split method over the system of equations to find the space-time Lagrange polynomial weights in tensor $\hat{\mathbf{w}}_{ij,\mathfrak{q}'}$. Finally, taking the product of $\mathbf{f}^*(\hat{\mathbf{w}}_{ij,\mathfrak{q}'})$ and its derivative, before integrating over space and time via the Gaussian quadrature scheme yields the values $\mathbf{F}_{ij}$ and $\mathbf{G}_{ij}$. [4, 28, 33]

3.4 The Osher-Solomon Riemann solver

The integrals $\mathbf{P}^{\pm}_{i+\frac{1}{2},j}$ and $\mathbf{Q}^{\pm}_{i,j+\frac{1}{2}}$ are approximated as a Gaussian quadrature sum at the interface between each element, using the linear phase path $\Phi = \mathbf{w}^- + s(\mathbf{w}^+ - \mathbf{w}^-)$. At each Gaussian point at the interface $x_{i+\frac{1}{2}}$. There are two predictions for $\hat{\mathbf{w}}$ produced by the left hand and right hand cells, giving a Riemann problem which can be solved to find (38) and (39). [27]

3.5 Source Term Splitting

Whilst the implicit solution nature of the DG predictor means that it can apply stiff source terms without the risk of numerical instability, the small number of implicit time steps taken (equal to the order of the approximation), means that the solution may be highly inaccurate. Instead, these source terms are applied via a splitting method, the BN equations are split as follows:

$$\bar{\mathbf{w}}^{n+1/2} = \bar{\mathbf{w}}^{n} - \frac{\Delta t}{\Delta x}[\mathbf{P}^{-}_{i+\frac{1}{2},j} + \mathbf{P}^{+}_{i-\frac{1}{2},j}] - \frac{\Delta t}{\Delta y}[\mathbf{Q}^{-}_{i,j+\frac{1}{2}} + \mathbf{Q}^{+}_{i,j-\frac{1}{2}}] - \Delta t(\bar{\mathbf{F}}_{i,j} + \bar{\mathbf{G}}_{i,j}) \quad (53)$$

with $\bar{\mathbf{w}}^{n+1/2}$ used as the initial condition to the source-split equation:

$$\bar{\mathbf{w}}^{n+1}_{ij} = \bar{\mathbf{w}}^{n+1/2}_{ij} + \int_{t^n}^{t^{n+1}} \mathbf{S}(x_{ij}, y_{ij}, t)dt \quad (54)$$

This integral is evaluated using an adaptive trapezium rule method, which reduces the size of the trapezium rule panel for large values of $\frac{\partial \mathbf{S}}{\partial \mathbf{w}}$

3.6 Stability Conditions

Generally numerical solutions for Hyperbolic PDE's require a time constraint to maintain stability, this often takes the form of:

$$\Delta t < C\frac{\Delta x}{S_{max}} \quad (55)$$

Where S_{max} is the maximum wave speed present in the solution and C is some stability constraint. Qualitatively this can be explained by the rate at which information physically travels through the system and the rate at which information can travel through the system within the model. If the physics of the problem allow information to travel faster than can be resolved by the model, information is destroyed and thus can cause the model to become unstable.

The upper bound for S_{max} can be found as the maximum Eigenvalue of the two flux matrices $\mathbf{F}(\mathbf{w})$ and $\mathbf{G}(\mathbf{w})$, given as:

$$|S_{max}| \leq max\{a_{k,ij} + |\mathbf{u}_{k,ij}|\} \quad (56)$$

Which can then be used as an estimate for S_{max}

A stability analysis of the discontinuous Galerkin predictor reveals that the stability constraint is given by:

$$C = \frac{1}{2(N-1)+1} \quad (57)$$

Where N is the order of the scheme [29]. Thus the choice of time step used within this work is given by:

$$\Delta t = C_{cfl} \quad \frac{\Delta x}{(2N-1)S_{max}} \tag{58}$$

With the choice of CFL condition number $C_{cfl} = 0.95$.

3.7 The Modified Ghost Fluid Method

The interface between a reflective solid boundary and a fluid is physically identical to that of a fluid interacting with its mirror image across the boundary. This property can therefore be used as a method for implementing moving solid boundary conditions is the following way. The problem can initially be solved over one time step without any geometric considerations, then the location of the fluid/solid interface can be found using the level set function described below. The zero of the level-set function describes the fluid solid interface , with the solid boundary conditions applied to the fluid as a reflection along this zero level-set. This is practically achieved by defining a region of 'ghost cells' within the solid region of the domain, where a secondary PDE is solved which interpolates velocities, pressure and density from across the interface. This process can be seen in the before and after images in Figure 4.

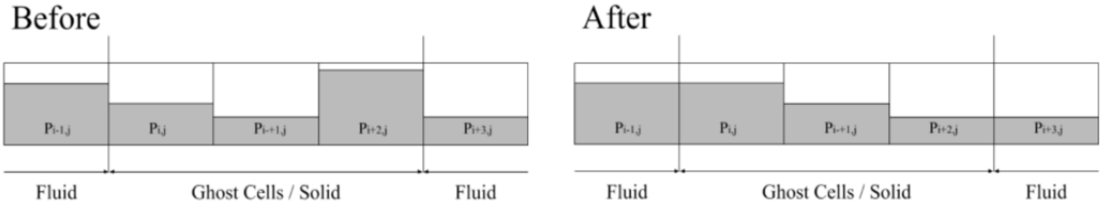


Figure 4: The ghost fluid method: the extrapolation equation is applied over the ghost cells, during the iteration, the two outermost ghost cells are given the values of pressure (and density, tangential velocity etc) from the adjacent fluid cells.

At every time step, n, of the solver (for dynamic structure problems), a signed distance / level set function, ϕ is calculate with magnitude equal to the smallest distance between a given cell and the fluid/structure interface. It is then given a sign depending on:

$$\phi(\mathbf{x}_{ij}) = \begin{cases} < 0 & \Omega_{ij} \in fluid \\ > 0 & \Omega_{ij} \in solid \end{cases} \tag{59}$$

where $\mathbf{x}_{ij}$ is the coordinate of each finite volume cell and the material interface is assigned to be at $\phi(\mathbf{x}) = 0$. The gradient of the level set function can then be used to find the unit normal to the solid/fluid boundary:

$$\hat{\mathbf{N}} =< n_x, n_y >= \frac{< \phi_x, \phi_y >}{| < \phi_x, \phi_y > |} \tag{60}$$

The interpolation PDE within the ghost fluid region is defined as:

$$\hat{\mathbf{N}} \cdot \nabla J = 0 \tag{61}$$

The solution of which can be found through solving the following Hyperbolic equation to steady state:

$$J_\tau + n_x J_x n_y J_y = 0 \tag{62}$$

Where J is any non-velocity variable being interpolated and subscripts τ,x and y are derivatives w.r.t imaginary time and real space. This can be discretized as

$$J^{k+1} = J^k - \frac{\Delta t}{\Delta x}[n_x J_x^\pm + n_y J_y^\pm] \tag{63}$$

Using upwinded first order finite difference approximations, Where K is the number of imaginary time steps taken and the superscript ± represents the up-winded derivative given by:

$$J^\pm_{x,(i,j)} = \begin{cases} \frac{J_{i,j} - Ji-1,j}{\Delta x} & n_x > 0 \\ \frac{J_{i+1,j} - Ji,j}{\Delta x} & n_x < 0 \\ 0 & n_x = 0 \end{cases} \tag{64}$$

The velocity condition takes the form of:

$$\mathbf{N} \nabla \cdot \mathbf{u} = \mathbf{0} \tag{65}$$

which can be discretized as a hyperbolic PDE in the same manor as:

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \frac{\Delta t}{\Delta x}\mathbf{N}[J_x^\pm + J_y^\pm] \tag{66}$$

This can be implemented through the following algorithm:

1. The relative velocity of the fluid to the solid is found through $\mathbf{u}_{relative} = \mathbf{u}_{fluid} - \mathbf{u}_{solid}$.
2. The vertical and horizontal reflected velocity components are calculated using the direction of the normal and the relative velocity.

$$\mathbf{u}_{reflected} = \mathbf{u}_{relative} - (\mathbf{u}_{relative} \cdot \mathbf{N}) \cdot \mathbf{N} \tag{67}$$

3. The velocity is transformed back into the original reference frame using $\mathbf{u}_{interpolate} = \mathbf{u}_{reflected} + \mathbf{u}_{solid}$, which can then be used in the original interpolation equation

The interpolation equation is then iterated to produce a steady state following each time step. This produces an initial condition for the next time step that enforces the solid boundary conditions. [24] [34] [35] [36] [37] [19]

3.8 Summary of the Complete Algorithm

The previous section describes in detail the numerical procedures required to numerically solve the BN equations in moving geometries. For completeness, the unified ADER-WENO BN algorithm is broadly described here. The total algorithm was implemented in the C++ programming language using the Armadillo [38] linear algebra package, written specifically for this application.

1. The vector of primitive variables is reconstructed from piecewise data using the WENO method to derive a spatial polynomial over each computational cell.

2. The DG predictor is used to produce polynomials describing describing the evolution of each variable over both space and time for each cell.

3. These polynomials are evaluated at each cell edge to produce a Riemann problem between left and right hand predicted states that varies over time

4. Each Riemann problem is solved using the Osher-Solomon method and integrated in time to calculate the intercell flux.

5. The intercell flux is used to update the piecewise data by one time step.

6. The source terms are then applied through integration over time.

7. The modified ghost fluid method is applied to reflect the fluid at any solid boundaries.

8. Boundary conditions are applied within the surrounding domain.

9. There is now an updated vector of piecewise constant data for each cell at time t^{n+1}

The complete algorithm was then validated through comparison to analytical, experimental and existing numerical results. These results are described and discuss within the next section.

4 Numerical Validation Results

In this section, the numerical results to a set of physical problems are presented, that were used to verify the various aspects of the code. Here verification means that the model is numerically correct, rather than physically correct, i.e. all aspects of the model perform as intended against benchmark tests. Following this section, experimental validation is performed to demonstrate the solver's ability to produce meaningful data. Initially, problems which test individual aspects of the solver are tested, before moving on to more complicated problems that test a combination of the various numerical aspects of the code.

4.1 Riemann Problem Tests

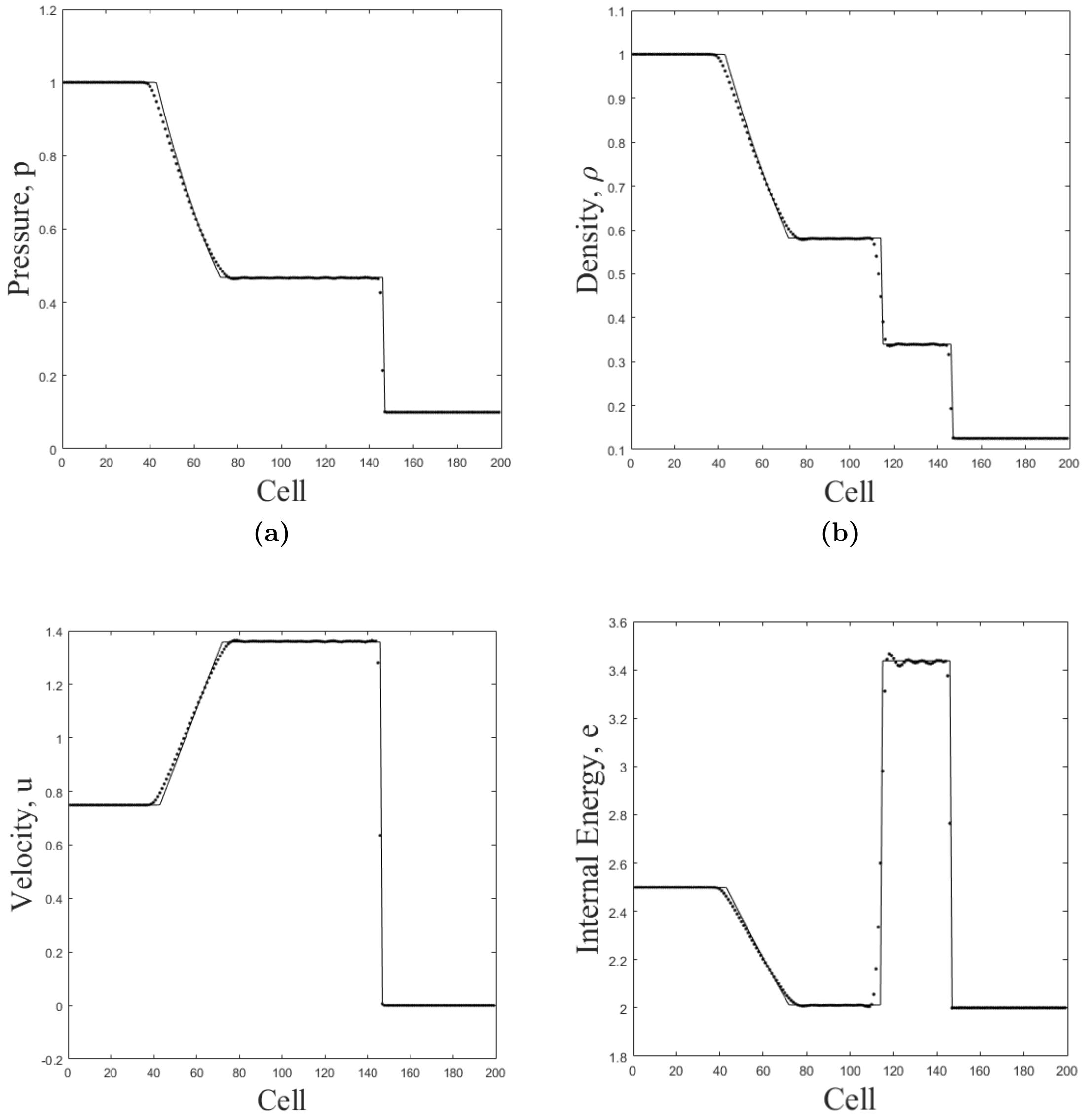
The ADER-WENO solver was initially tested against the 5 standard Riemann problem tests found in [25]. These are 1D single phase tests that allow the solution framework to be checked against an analytical result and provide an initial 'sanity check' for the method. Following this, the two phase Baer-Nunziato Riemann problems were simulated in the full BN ADER WENO solver and compared against similar analytical results. The following sections describe the problem setups before discussing and analyzing the results.

Single Phase Euler equations

These tests involve simulating a 1D domain of length $l = 1.0$ with a discontinuity in the fluid properties at a point x_0 distance from the origin. The system is then allowed to evolve in time before reaching a final state at t_{end}. The ADER-WENO solver was run using 200 computational cells with the DG error threshold set to $\epsilon_{DG} = 1 \times 10^{-12}$.

Test	ρ_L	u_L	p_L	ρ_R	u_R	p_R	x_0	t_{end}
1	1.0	0.75	1.0	0.125	0.0	0.1	0.3	0.2
2	1.0	-2.0	0.4	1.0	2.0	0.4	0.5	0.15
3	1.0	0.0	1000.0	1.0	0.0	0.01	0.5	0.035
4	5.9992	19.598	460.89	5.99242	-6.19633	46.0950	0.4	0.035
5	1.0	-19.5975	1000.0	1.0	-19.59745	0.01	0.8	0.012

Table 1: Initial conditions for the five Riemann problem tests of Toro [3], in all cases an ideal gas equation of state was used with $\gamma = 1.4$



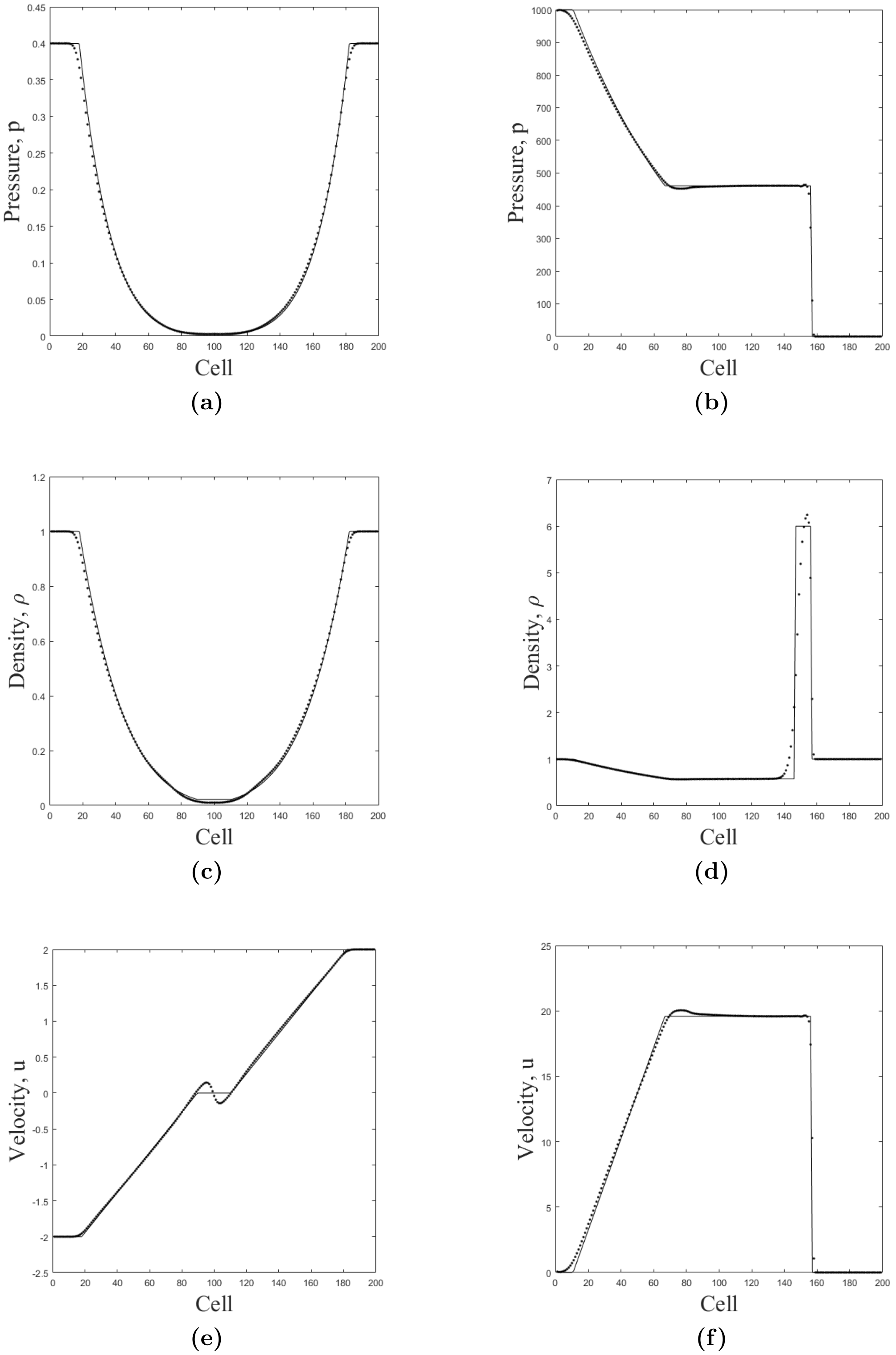
(a)

(b)

(c)

(d)

Figure 5: Plots of (a) pressure, (b) density, (c) velocity and (d) internal energy for Toro's first Riemann problem test with exact solution in black and 3rd order ADER-WENO solution as points.

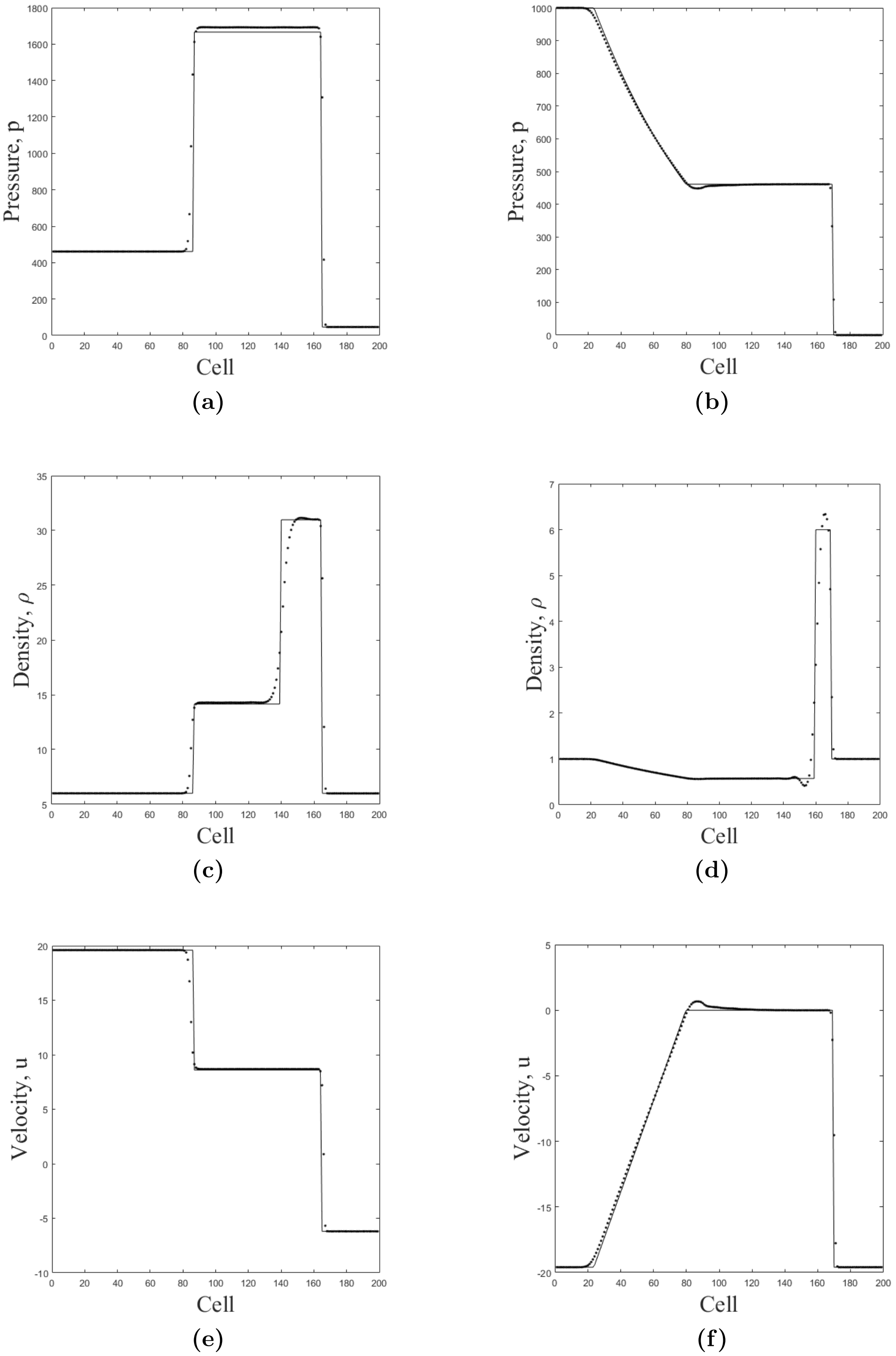


(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 6: Plots of (a) pressure, (c) density, (e) velocity and (d) internal energy for Toro's second Riemann problem test and plots of (b) pressure, (d) density (f) velocity for Toro's third Riemann problem test. The solid line is the exact solution, whilst the dotted line is the third order ADER-WENO numerical result.



(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 7: Plots of (a) pressure, (c) density, (e) velocity and (d) internal energy for Toro's fourth Riemann problem test and plots of (b) pressure, (d) density (f) velocity for Toro's fifth Riemann problem test. The solid line is the exact solution, whilst the dotted line is the third order ADER-WENO numerical result.

Multiphase Baer-Nunziato equations

The solver was modified to accompany the two phase Baer-Nunziato equations and run against the Riemann-type problems found in Dumbser [39]. All tests were run to a final time of $t_{end} = 0.1$ using 200 computational cells on a total domain length of $L = 1$, with the discontinuity located at $x = 0.5$. The DG error threshold was set to $\epsilon_{DG} = 1 \times 10^{-14}$

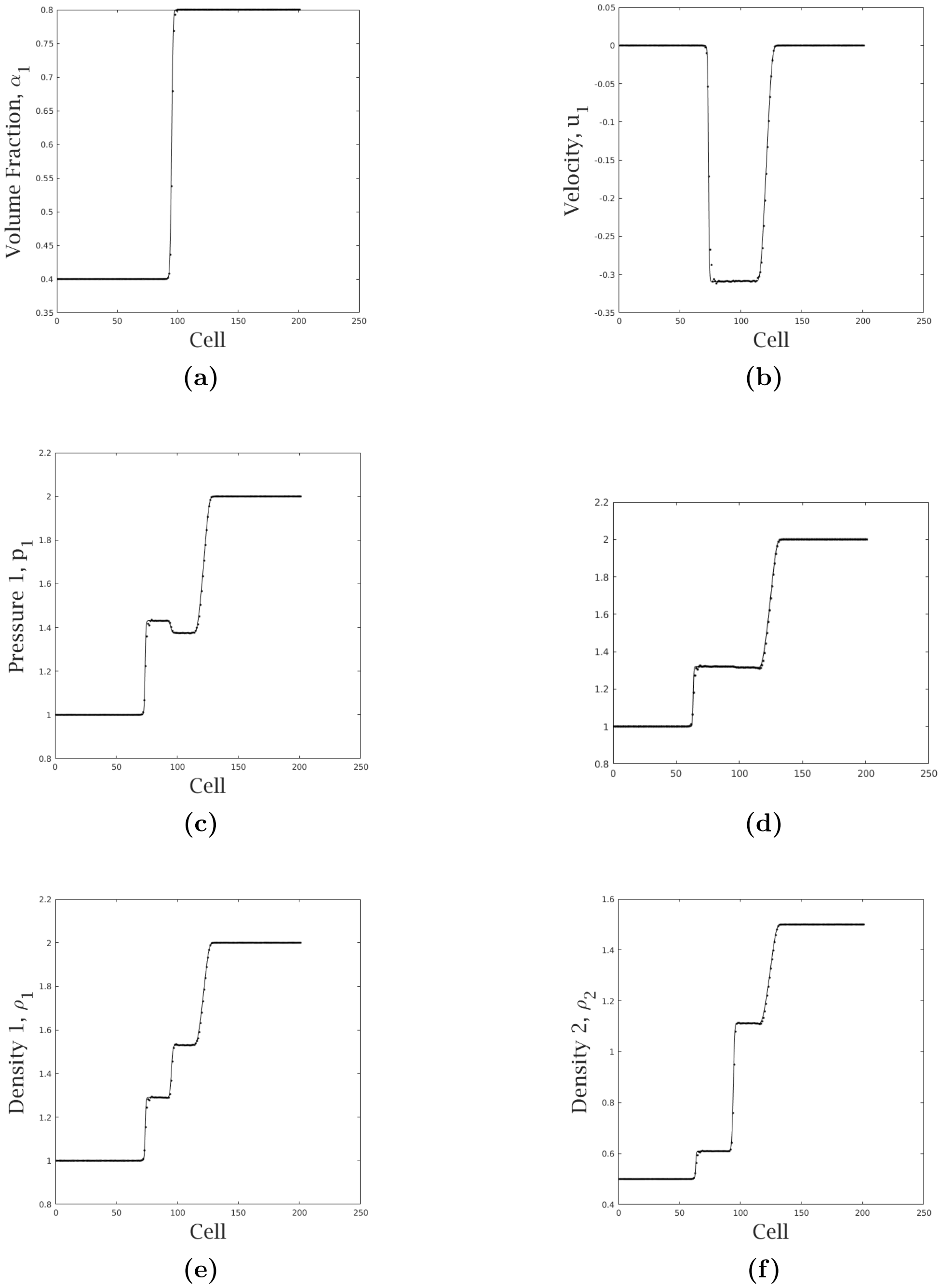
Test	state	α_1	ρ_1	u_1	p_1	ρ_2	u_2	p_2	γ_1	π_1	γ_2	π_2
1	L	0.4	1.0	0.0	1.0	0.5	0.0	1.0	1.4	0.0	1.4	0.0
	R	0.8	2.0	0.0	2.0	1.5	0.0	2.0				
2	L	0.4	800.0	0.0	500.0	1.5	0	2.0	3.0	100.0	1.4	0.0
	R	0.3	1000.0	0.0	600.0	1.0	0.0	1.0				
3	L	0.2	1.0	0.9	2.5	1.0	0.0	1.0	1.4	0.0	1.4	0.0
	R	0.9	1.0	0.0	1.0	1.2	1.0	2.0				

Table 2: Initial conditions for the three BN Riemann problem tests of Dumbser et al. [4]

Analysis

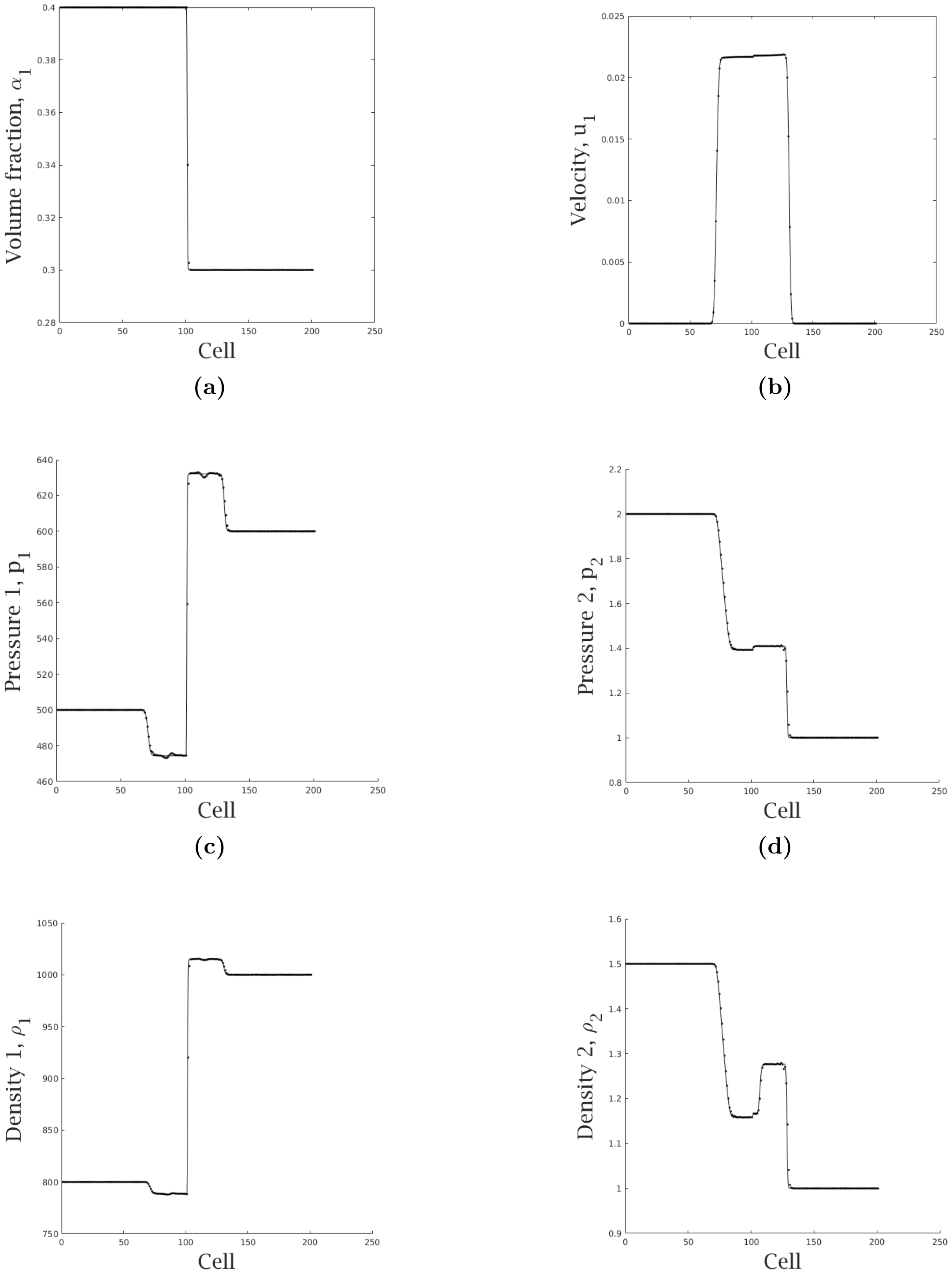
From Figures 5-10, it can be seen that the 3rd order ADER-WENO solver produced numerical results that very closely matched the exact solution for both the single phase Euler equations and the two phase BN equations. In all cases, the numerical solution accurately tracked the shock-wave propagation speed and resolved large discontinuities within the pressure, density and velocity profiles with minimal numerical diffusion. However, despite using the WENO method for actively reducing oscillations within the piecewise reconstruction, the high order scheme did produce some numerical dispersion, which can be more clearly seen in Figures 6(e), 7(d) and 10(b). This can be further reduced by decreasing the error threshold, however this can add significant computational overhead and may not be feasible moving into more complicated 2D problems.

Additionally it was found that without drastically decreasing the size of the error threshold, the the solver would work up to an order of accuracy of 3 using 200 cells. Beyond this, more significant numerical dispersion would be produced and often cause the scheme to become unstable. Additionally, the 2D ADER-WENO algorithm scales as $O(2^{2N-1}N^3)$ - (2 spatial and 1 temporal dimension to resolve combined with the CFL condition), meaning a fourth order scheme would take about 10 times longer than the third order for a limited increase in accuracy.



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 8: Plots of (a) Phase 1 volume fraction, (b) Phase 1 velocity, (c) phase 1 pressure, (d) Phase 2 pressure, (e) Phase 1 density, (f) Phase 2 density for the first Baer-Nunziato Riemann problem at time $t = 0.1$

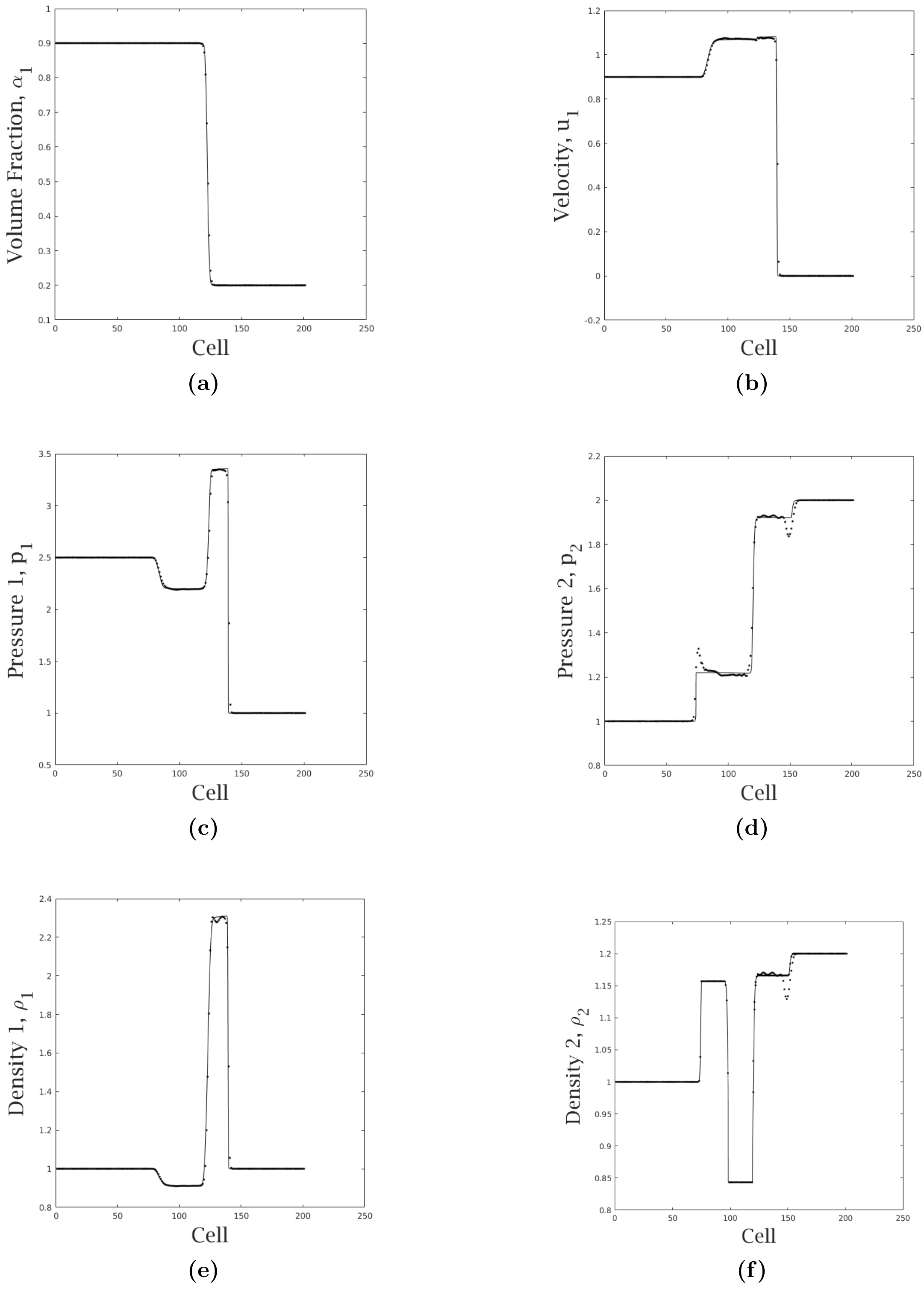


(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 9: Plots of (a) Phase 1 volume fraction, (b) Phase 1 velocity, (c) phase 1 pressure, (d) Phase 2 pressure, (e) Phase 1 density, (f) Phase 2 density for the second Baer-Nunziato Riemann problem at time $t = 0.1$



(a)

(b)

(c)

(d)

(e)

(f)

Figure 10: Plots of (a) Phase 1 volume fraction, (b) Phase 1 velocity, (c) phase 1 pressure, (d) Phase 2 pressure, (e) Phase 1 density, (f) Phase 2 density for the third Baer-Nunziato Riemann problem at time $t = 0.1$

4.2 ADER-WENO Error Analysis

Here , two simple problems are examined in order to qualitatively examine the discretization errors inherent in the ADER-WENO scheme by looking at the effects of grid-resolution and the order of the scheme.

Effect of Resolution

In this section the 3rd order ADER-WENO scheme was used to simulate a smooth compressible problem at various grid resolutions in order to compare error to grid resolution. The problem setup was as follows, initially the single phase fluid had initial conditions $[\rho, u, v, p] = [1, 0, 0, 1 + exp(-100(x - 0.5)^2)]$ with pass-through boundary conditions at either end of the domain. The system was then allowed to evolve to a final state at $t = 0.15$ and its error norm recorded.

To estimate the error norm of each simulation, a high resolution simulation with 2000 cells was initially run and used to produce a base solution to which other simulations could be compared to. Following this, several lower resolution simulations were run and the root mean squared error norm between the high resolution and lower resolution pressures were recorded. A log-log plot of error norm against resolution can be seen in Figure 11. The line of best fit produced has a gradient of -2.1, implying that the accumulated error has a quadratic convergence, $O(\Delta x^2)$.

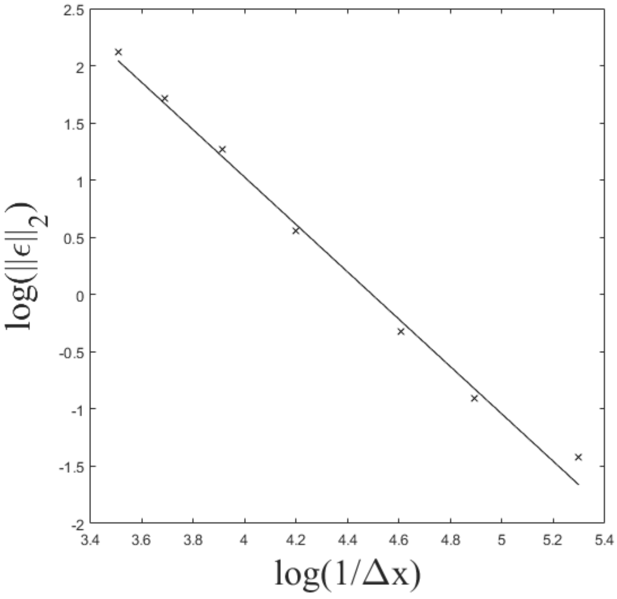


Figure 11: Log-log plots of error norm against resolution for the evolution of a Gaussian pressure distribution

Effect of Scheme Order

The effect of resolution on a Riemann-type problem can be seen in Figure 12. The first order scheme shows significant smoothing of discontinuities, as does the second order scheme to a lesser extent. The third order scheme matches the exact solution extremely well, however some oscillations start to become present in the region where two discontinuities are nearby, for higher order schemes, these oscillations start to make the scheme become numerically unstable, due to Runge's phenomenon [35].

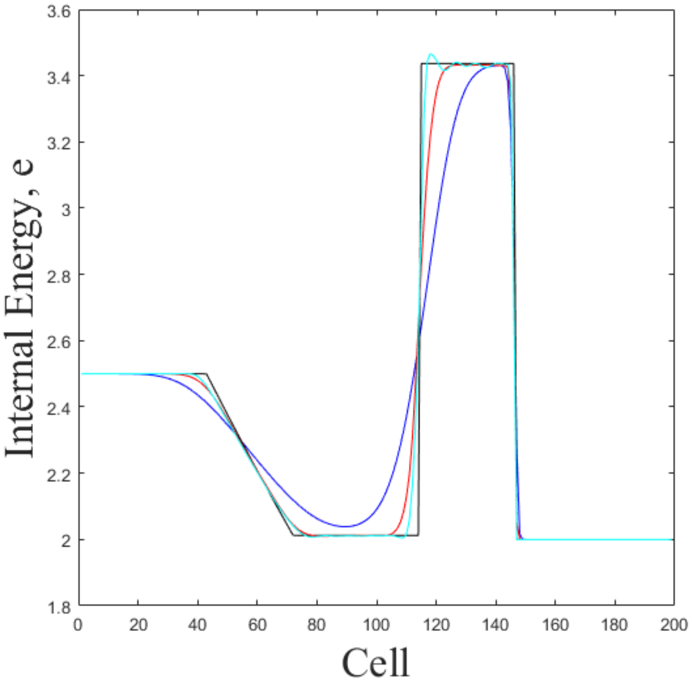


Figure 12: Plots of internal energy for Toro's first Riemann problem for a first order ADER-WENO scheme (blue), a second order ADER-WENO (red) a third order (cyan) and the exact solution (black).

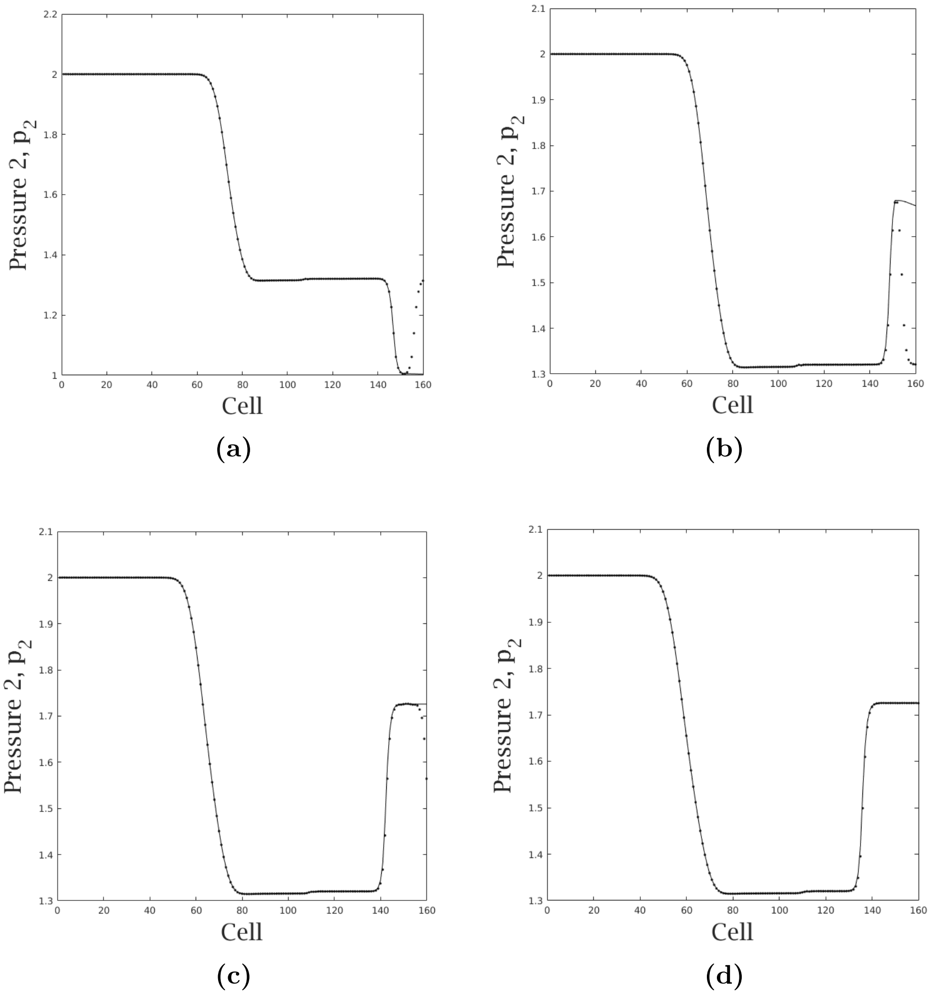
4.3 Fixed Ghost Fluid Boundary test

The MGFM essentially models solid boundaries as a reflection normal to the surface of the solid material. Therefore a simulation of a collision between two identical shock waves should produce the same result as a single shock wave colliding with a ghost fluid solid boundary. The initial setup is shown in Table 3 on a domain of length $L = 2.0$. Here L refers to the state left of the discontinuity $x_L = 0.5$, C stands for the state in between the discontinuities $x_L = 0.5$ and $x_R = 1.5$ and R is the state to the right of $x_R = 1.5$, all distance are taken from the origin. This is essentially a repeat of the first Baer-Nunziato Riemann problem test, mirrored at $x = 0.25$.

State	ϕ_1	ρ_1	u_1	v_1	p_1	ρ_2	u_2	v_2	p_2
L	0.75	1999.939402	49.998485	0	4999849.5	1.0	0	0	1.0
C	0.75	1000	0	0	1.0	1.0	0	0	1.0
R	0.25	1000	0	0	1.0	1.0	0	0	1.0

Table 3: Initial conditions for two phase shock collision with EOS parameters: $\gamma_1 = 3.0$, $\pi_1 = 100$, $\gamma_2 = 1.4$, $\pi_2 = 0$

.



(a)

(b)

(c)

(d)

Figure 13: Secondary phase pressure plots of the shock wave collision with a reflective wall using the GFM (solid line) and two shock wave collision (points) at (a) $t = 0.1$, (b) $t = 0.2$, (c) $t = 0.3$ (d) $t = 0.4$.

Starting at $t = 0$, the left hand shock wave propagates rightward toward the ghost fluid/opposing shock wave. Once the two meet, there is a rapid increase in pressure directly adjacent to the wall as the velocity profile is reflected back. This generates a secondary leftward moving shockwave which then propagates through the domain. Figure 13 shows that the two methods produce almost identical results with negligible error between them.

4.4 The 1D compaction problem

This section examines the 1D shock-tube type granular compaction problem first modeled by Schwendeman et al. [5]. All units are given in dimensionless form to accurately match the conditions used in the original study.

Initially at $t = 0$, a shock tube containing 73% by volume fraction of a granular solid is at rest relative to the shock tube's frame of reference. A piston then travels at $u_p = 0.013318$ rightward through the domain, generating a rightward traveling compaction wave with velocity $u_c = 0.038$ relative to the piston. As the compaction wave propagates, it locally increases the solid volume fraction, total pressure and total density - as one would expect for compacted granular media. The problem was modeled both in the frame of reference of the piston and the shock tube and the results compared in Figures 14-15. The initial setup for the shock tube frame of reference model can be found in Table 4 along with the equation of state parameters.

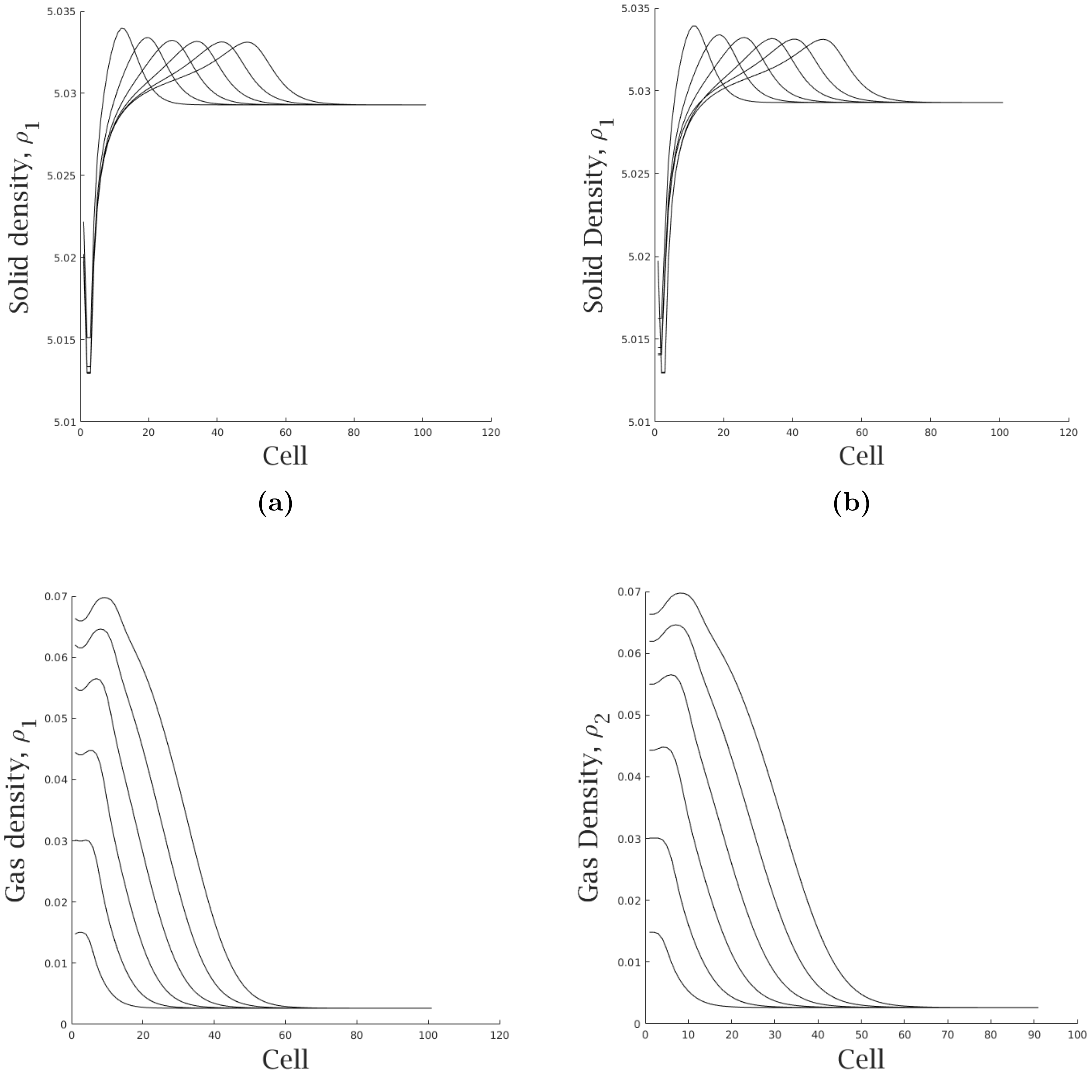
Solid	Value	Gas	Value	Solid EOS	Value	Gas EOS	Value
α_1	0.73	α_2	0.27	γ_1	5	γ_2	1.35
ρ_1	5.0293	ρ_2	$2.6470x10^{-3}$	π_1	0.16021	b_2	0.37778
p_1	3.5681×10^{-4}	p_2	1.1843×10^{-5}	C_v^1	0.625	C_v^2	1
u_1	-0.013318	u_2	-0.013318	q_1	0.11795		

Table 4: Initial conditions and equation of state parameters for the 1D inert granular compaction problem found in [5], a stiffened gas equation of state was used for the solid phase, whilst a Virial equation of state was used for the surrounding phase.

In this setup, the boundaries of the domain were allowed to remain fixed. A reflective boundary condition was used on the left-hand boundary to simulate the moving piston and a pass-through condition at the right-hand boundary so as to not introduce additional compaction/dispersion waves. In the case of the shock-tube frame of reference, a moving ghost fluid boundary was initially place at x=0 and traveled rightward through the domain. The initial setup was identical to Table 4, however with the initial velocity set to zero. The graphs in Figures 14-15 are taken in the frame of reference of the piston for easier comparison.

The two simulations show a strong concordance with the results of Schwendeman et al. The compaction wave in both cases travel at the same rate as in Schwendeman, $u_c = 0.038$ and produce the same maximum solid volume fraction limit of around 0.98 at the piston face. Characteristically, the two models also produce similar density, velocity and pressure profiles, with near identical peak values for example a maximum solid density value of 5.035 compared to 5.037 in Schwendeman.

However, there are some numerical artifacts present within the solution. The two density plots show a clear local reduction in density near the piston face at the left-hand boundary. Unlike the plots of volume fraction and pressure which also demonstrate small dips, the reduction in pressure is not contained within the boundary region, it is actually present within the numerical solution. This phenomena is often referred to as the 'wall-heating problem' due to the way it artificially raises the temperature at boundaries. Within detonation studies, this can cause premature ignition of fuel and thus produce highly incorrect results. However in the context of this inert phase work, this boundary problem can mostly be ignored.



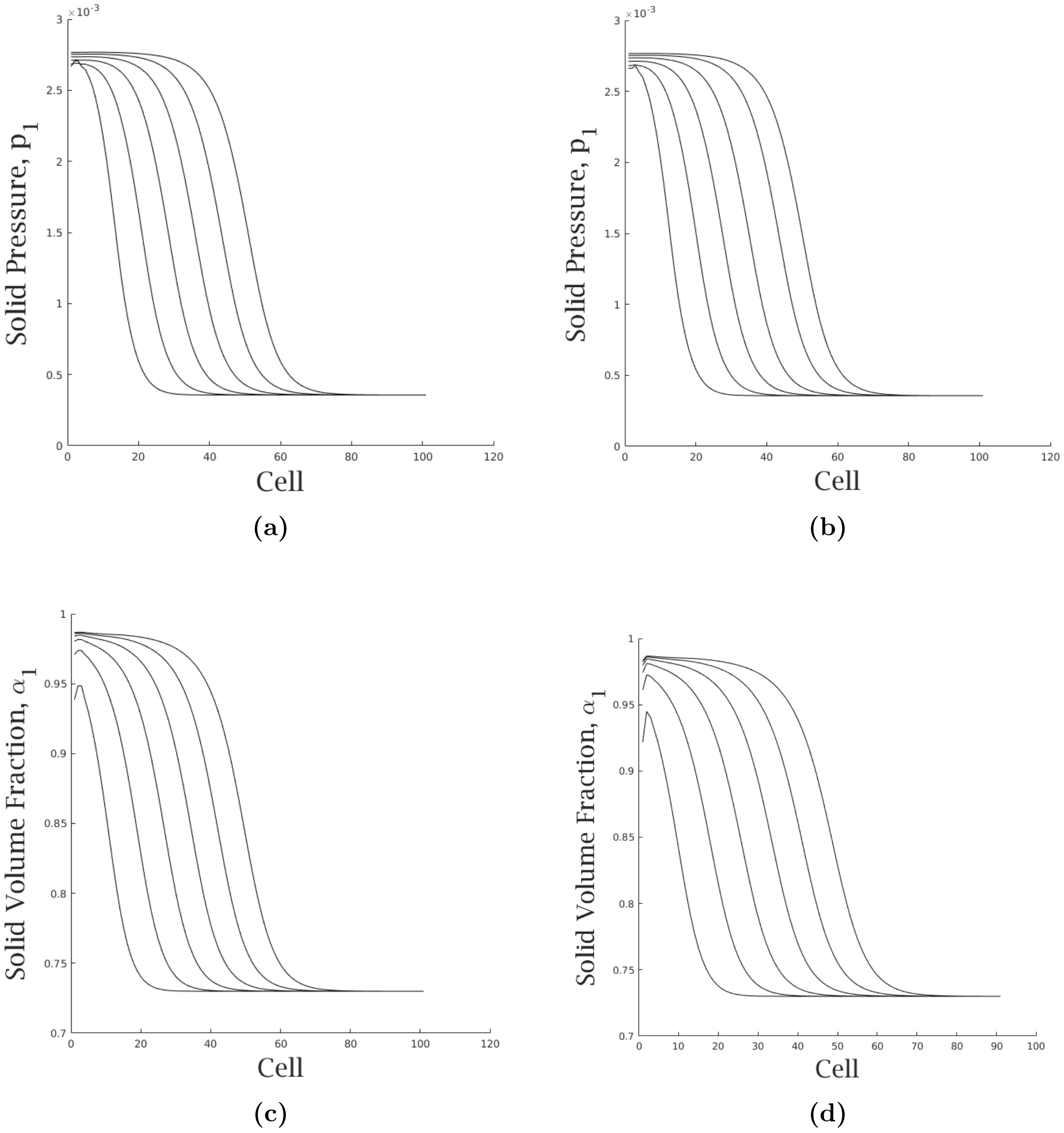
(a)

(b)

(c)

(d)

Figure 14: Plots of the evolution of solid density over time for (a) standard wall boundary conditions (b) the moving ghost fluid conditions and the evolution of gas density for (c) standard wall boundary conditions and (d) the moving ghost fluid condition for times $t = \{50 : 50 : 300\}$



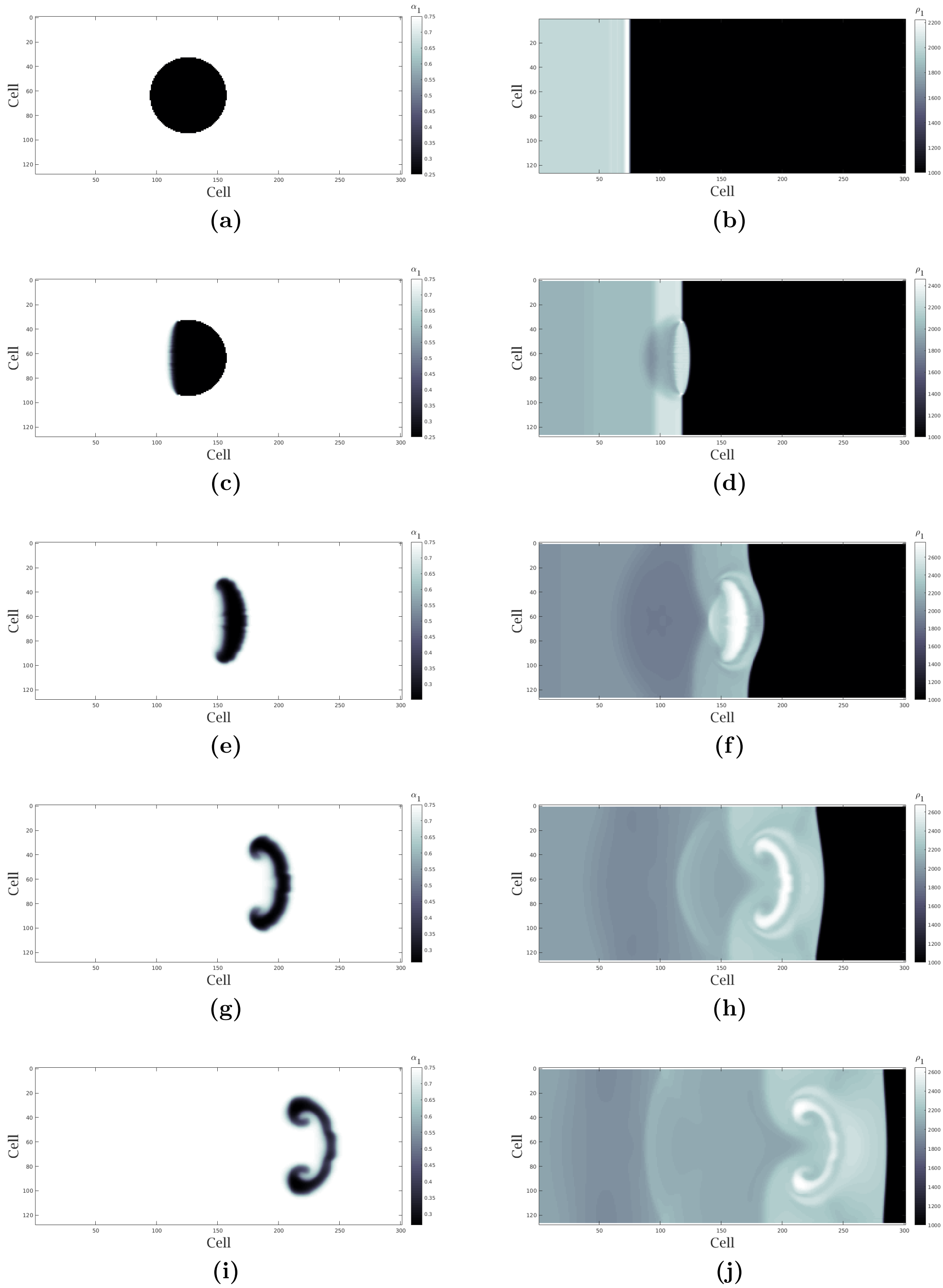
(a) **(b)** **(c)** **(d)**

Figure 15: Plots of the evolution of solid pressure over time for (a) standard wall boundary conditions (b) the moving ghost fluid conditions and the evolution of solid volume fraction for (c) standard wall boundary conditions and (d) the moving ghost fluid condition for times $t = \{50 : 50 : 300\}$

4.5 Bubble/shock-wave interaction

In this section, a simplified shock-bubble interaction problem, first examined by Dumbser [39], is simulated as a benchmark test for the multiphase solver.

A bubble of radius $R = 0.25$ is centered at $x = 1$, $y = 0.5$ in a 2.5×1 unit length domain. The initial conditions are given by a Riemann problem shown in Table 5, with the initial discontinuity located at $x = 0.5$.



(a) **(b)**

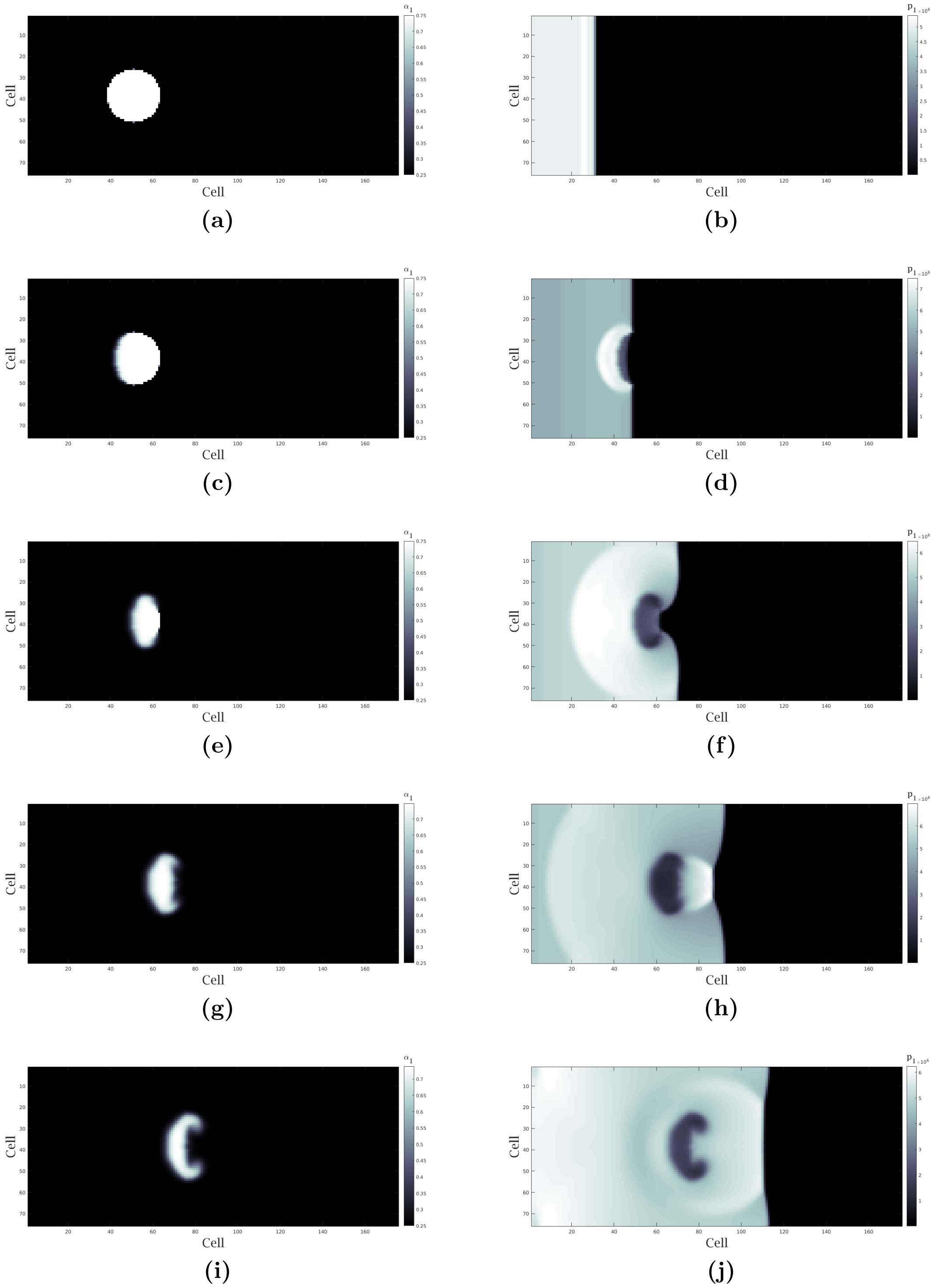
(c) **(d)**

(e) **(f)**

(g) **(h)**

(i) **(j)**

Figure 16: Plots of liquid volume fraction for a liquid shock wave interaction with an air bubble at (a) $t = 0.005$, (c) $t = 0.010$, (e) $t = 0.015$, (g) $t = 0.020$, (i) $t = 0.025$ and liquid pressure at (b) $t = 0.005$, (d) $t = 0.010$, (f) $t = 0.015$, (h) $t = 0.020$, (j) $t = 0.025$.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 17: Plots of liquid volume fraction for a gas shock wave interaction with a liquid bubble at (a) $t = 0.005$, (c) $t = 0.010$, (e) $t = 0.015$, (g) $t = 0.020$, (i) $t = 0.025$ and liquid pressure at (b) $t = 0.005$, (d) $t = 0.010$, (f) $t = 0.015$, (h) $t = 0.020$, (j) $t = 0.025$.

State	ϕ_1	ρ_1	u_1	v_1	p_1	ρ_2	u_2	v_2	p_2
L	0.75	1999.939402	49.998485	0	4999849.5	1.0	0	0	1.0
R	0.75	1000	0	0	1.0	1.0	0	0	1.0
Bubble	0.25	1000	0	0	1.0	1.0	0	0	1.0

Table 5: Initial conditions for the bubble / shock wave interaction problem, a stiffened gas equation of state was used for both phases with parameters: $\gamma_1 = 3.0$, $\pi_1 = 100$, $\gamma_2 = 1.4$, $\pi_2 = 0$.

The two initial states are given by the Rankine-Hugoniot conditions for a speed S=100 shock wave, which then propagates rightward through the domain, colliding with the bubble. The evolution of the system can be seen in Figures 16 and 17 where the volume fraction and pressures are shown up to a final time of $t = 0.02$. Initially the bubble is compressed with an increase in both Phase 1 volume fraction and pressure within the bubble. The high pressure within the bubble begins to dissipate, generating a secondary, radial shock wave, seen in Figure 16(f). This secondary shock wave elongates the shape of the bubble into a crescent like shape, seen in the final plot in Figure 17.

Additionally, the corners of the bubble begin to curl into vortex-like forms - acting to mix the two phases. This phenomena is caused by the sudden shock-wave acceleration of the two bubble phases, producing a Richtmyer-Meshkov (RM) type instability within the bubble. In this particular case (of a denser fluid accelerating a lower density fluid), the RM instability should act to elongate the lower density fluid as observed in this numerical study.

The Results shown here are almost identical to those produced by Dumbser up to $t = 0.015$, at this point the vortices at the corners of the bubble become significantly diffused compared to Dumbser. This is due to the lack of the adaptive mesh in this study compared to Dumbser's work and thus there is a lower resolution here than required to accurately simulate the thin vortex. This is discussed further in Section 7.

In the secondary study, the phases are reversed, such that the bubble contains a higher volume fraction of stiffened phase 1 than its surroundings. Interestingly, the top and bottom corners of the bubble travel at a faster rate than the center, opposite to the previous simulation and the bubble remains much squatter, with no sign of vortex formation. This behavior may be caused by a similar RM instability, but where the two roles are reversed with a denser fluid being penetrated by a lighter one. Theoretically this type of RM instability should produce bubbles that act to mix the two phases, but the lack of surface tension within this model combined with the low resolution mean that this phenomena is not observed within the model.

5 Experimental Validation Results

The results presented in Section 4 demonstrate the the ADER-WENO solver's ability to accurately model the Baer-Nunziato equations in complex geometries through comparison with analytical solutions, existing numerical work and computational predictions. Whilst the developed model is numerically accurate, this does not necessarily mean that the model is completely consistent with the physics of granular multiphase flows. The model must also be compared to experimental results.

Numerical simulations of two multiphase granular flow problems are presented within this section, for which experimental data exists: a 1D shock tube type problem and a slow granular flow around a cylinder, approximately modeled using a higher Mach number flow and modified compaction parameter to decrease required computation time at the expense of increased compressibility (still negligible).

5.1 Granular Shock Dispersion

The multiphase granular flow model is now compared to existing experimental data of a granular shock, initially captured by Rogue et al. [30], containing air at atmospheric pressure. The driver gas forms a shock wave of strength $Ma = 1.3$ and propagates through the shock tube. The initial problem setup can be seen in the diagram in Figure 18.

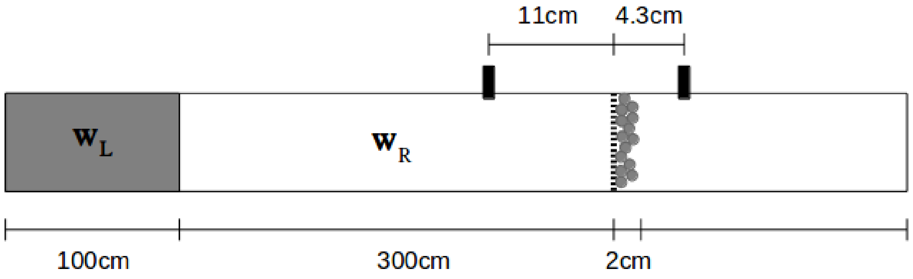


Figure 18: Diagram of the Shock-dispersion experimental setup with high pressure gas in the grey region marked $\mathbf{w}_L$, atmospheric pressure gas in the region marked $\mathbf{W}_R$ and a support structure containing a bed of particles shown as a dotted line. Two pressure gauges, downstream and upstream are shown as thick black markers.

The shock wave eventually collides with the particle bed, transferring momentum and energy to the particles and driving them away from the support. This action produces three secondary shock waves, a high pressure, fast moving reflected gaseous shock wave, a low pressure transmitted shock wave and a slow moving, high pressure, slow moving gas/particle cloud. The evolution of pressure is measured at two locations, 11cm below/downstream of the support and 4.3cm above/upstream of the support.

Rogue et al. demonstrated the presence of a weak compaction wave acting to disperse the particles, indicating that the packing energy has a negligible effect on the compaction rate. Thus the problem is modeled using Saurel's model with a packing energy gradient of $\beta = 0$ and a large pressure relaxation parameter of $\mu_F = 100$ to negate any induced surface tension effects. The initial condition data and model parameters are shown in Table 6:

State	ϕ_1	ρ_1	u_1	v_1	p_1	ρ_2	u_2	v_2	p_2
L	10^{-5}	2800	0.0	0.0	0.350×10^6	4.0648	0.0	0.0	0.350×10^6
R	10^{-5}	2800	0.0	0.0	0.101×10^6	1.177	0.0	0.0	0.101×10^6
Bed	0.65	2800	0.0	0.0	0.101×10^6	1.177	0.0	0.0	0.101×10^6

Table 6: Initial conditions for the granular dispersion problem with equations of state $\gamma_1 = 5.0$, $\pi_1 = 341.3 \times 10^6$, $\gamma_2 = 1.4$, $\pi_2 = 0$

.

Experimentally measured values of pressure evolution are shown in Figure 19, along with the numerical results produced by our model shown in Figure 20. The numerical results match the experimental model almost exactly with correct predictions for reflected pressure, refracted shock wave speed and particle/gas cloud movement speed. Qualitatively, the two results also match extremely well. The downstream pressure gauge initially shows a jump in pressure due to the primary Mach 1.3 shock wave, followed by a secondary jump from the reflected shock wave. This jump in pressure slowly decays as the reflected shock propagates backward through the domain. The upstream pressure gauge has a significantly smaller initial jump in pressure as the transmitted shock wave passes by, the pressure then slowly increases towards the reflected shock wave pressure as the gas/particle cloud moves through the domain.

Generally, the numerical model predicts slightly larger values of pressure at the two pressure gauges than observed in the experimental data, similar to the issues discussed in Saurel [2], caused by the thin geometry of the shock tube. These factors are ignored within this work due to the assumption of a 1D geometry. Overall however, the numerical model and experimental data match extremely well, verifying the physicality of our model.

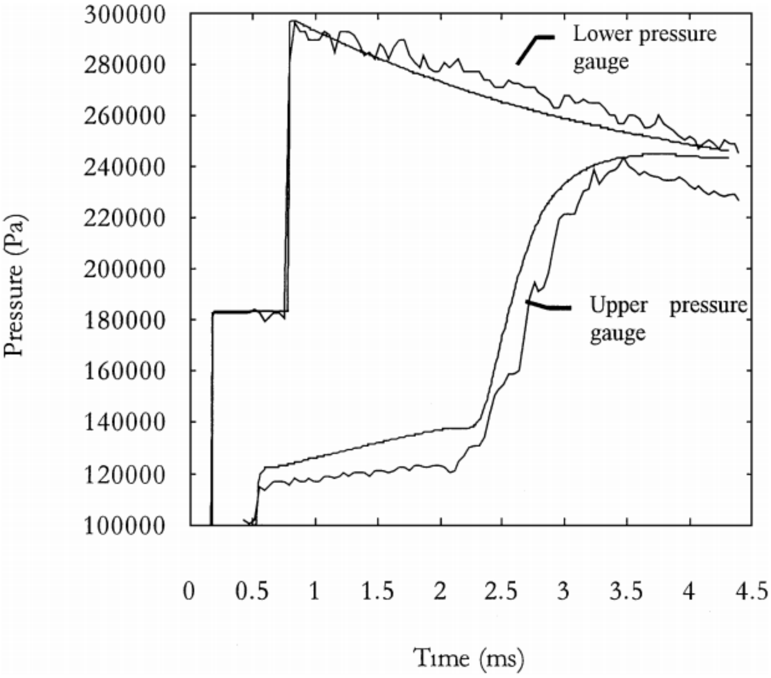


Figure 19: Experimental data of Gas pressure against time, taken from [2], showing the evolution of pressure at the upstream and downstream pressure gauges for a Mach 1.3 shock interacting with a 2cm thick nylon particle bed, the setup for which is shown in Table 6

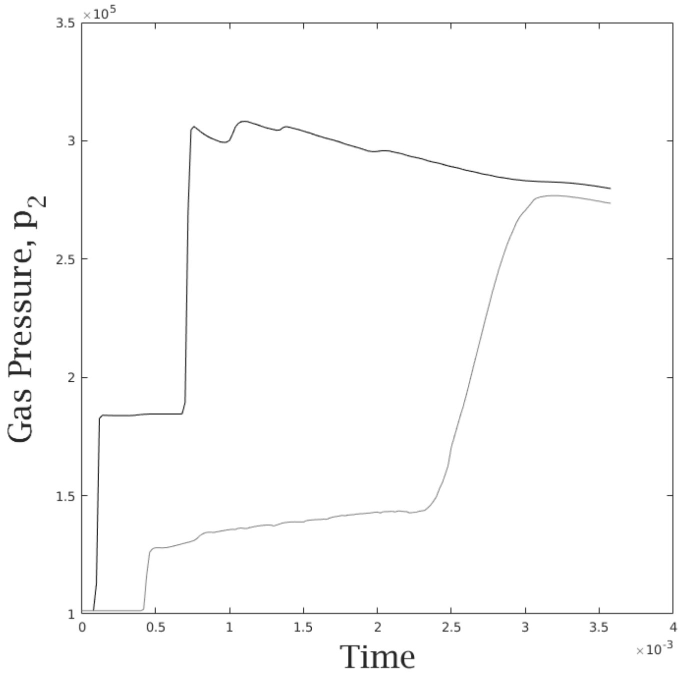


Figure 20: Numerical calculation of Gas pressure against time using compaction parameter 1. The evolution of pressure at the upstream and downstream pressure gauges for a Mach 1.3 shock interacting with a 2cm thick nylon particle bed is shown.

5.2 Weak Granular compaction around a Cylinder

This problem was originally presented by Chehata et al. [1] and demonstrates the effects of a slow, negligible Mach number granular flow around a cylinder. The experimental setup is as follows. A granular flow with particle size $d_p = 3mm$, volume fraction $\alpha_1 = 0.65$ and background velocity $U_p = 336mm/s$ is induced around a cylinder of diameter 38.1mm. The flow is allowed to evolve to a steady state, producing a teardrop shaped region of cavitation behind the cylinder and an area of compaction in front of it. A PIV method was then used to capture the flow velocity profiles, the results of which are numerically recreated here. The experimental setup can be seen in Figure 21.

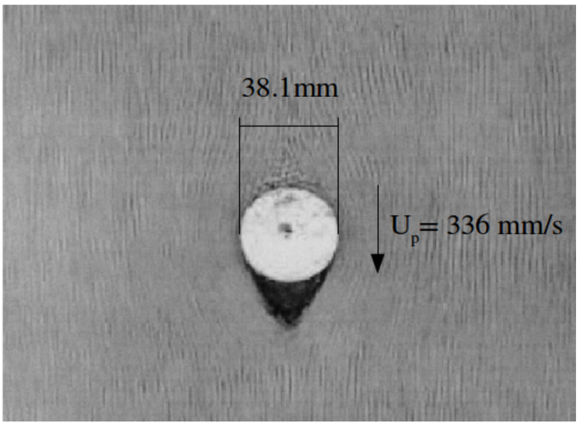


Figure 21: Image of the experimental setup for the weak granular compaction around a cylinder taken from [1]. The cylinder is centered within the domain with diameter 38.1mm with a background flow velocity of $U_p = 336mm/s$.

The low Mach number nature of the flow is extremely difficult to numerically resolve within the ADER-WENO MGFM framework developed within this work, due to two computational considerations. Firstly, compressible/compressible solvers generally have an error inversely proportional to Mach number, thus the negligible Mach number flow introduces numerical artifacts into the solution, seen as oscillations in the velocity profile in Figure 23. Secondly the CFL condition imposes an extremely small time-step relative to the speed of the flow development. Practically, this means that the time required to run the simulation is very large and it would therefore be infeasible to simulate the exact flow described here. Instead, a larger background velocity is used to enforce a faster flow development, $U_p = 33.6m/s$. This is a significantly larger Mach number, but is still small enough such that the compressible effects of the flow are negligible. The weak compaction model of Saurel is used to model the weak compaction waves present, with a compaction viscosity of $\mu_F = 10$ to negate the effects of using a higher velocity.

The experimental results from Cehata et al. [1] are shown in Figure 22, whilst the calculated numerical results are shown in Figure 23. The equation of state and compaction properties used were identical to the problem described in section 5.1.

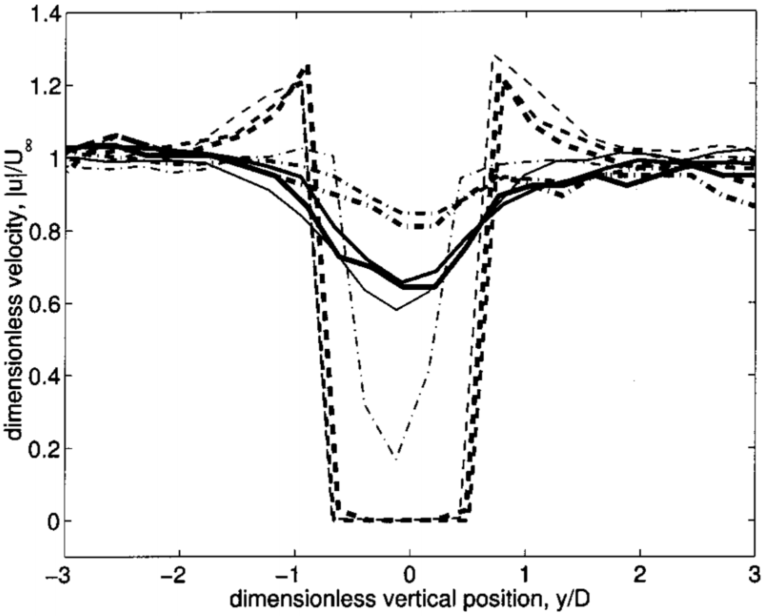


Figure 22: Experimentally found values of velocity magnitude taken from [1]. The cut-through locations are $y/D = -1.1$ (solid line), $y/D = 0.0$ (dashed line) and $y/D = 1.1$ (dot-dashed line) from the horizontal centerline of the cylinder.

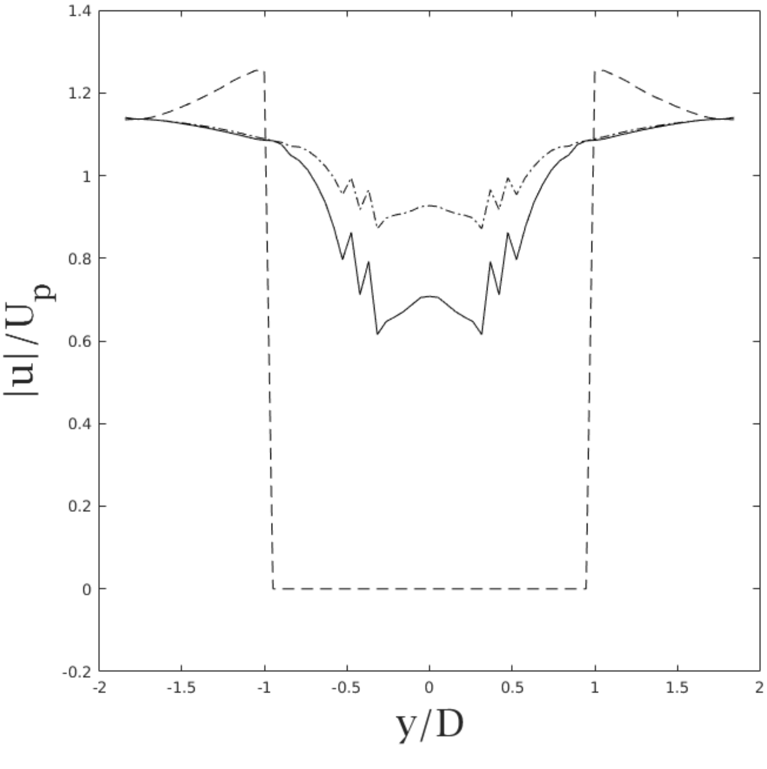


Figure 23: Numerically calculated values of velocity magnitude around the central portion of the flow domain at distances $y/D = -1.1$ (solid line), $y/D = 0.0$ (dashed line) and $y/D = 1.1$ (dot-dashed line) from the horizontal centerline of the cylinder.

The two results demonstrate a strong concordance with one another, with similar maximum velocity magnitudes of $|u|/U = 1.3$ at $y/D = 0.0$, minimum values of $|u|/U = 0.7$ at $y/D = -1.1$. The two results are also qualitatively similar producing an increased velocity magnitude at $y/D = 0.0$, decreased velocity magnitude profile at $y/D = 1.1$ and significant decrease in velocity profile at $y/D = -1.1$. There are however, clear discrepancies between the two. Large oscillation can be seen in both $y/D = 1.1$ and $y/D = -1.1$ caused by unstable behavior in the MGFM algorithm due to the low Mach number and low resolution (this is a well observed phenomenon [40]). The numerically calculated average velocity magnitude around the cylinder is also significantly higher than found in the experiment.

The differences between the numerical and experimental results can be mainly attributed to the difficulties described above, in simulating low Mach number flows with a compressible solver. This indicates that further work could be undertaken to make the solver more compatible with slower granular flows. However, as mentioned in section 1.2 these types of flow fall outside the algorithm specification, but can still be reasonably well resolved in spite of this.

6 Novel Results

In this final results section, two novel numerical results are presented to demonstrate the capabilities of the ADER-WENO MGFM solver in modeling two dimensional, moving geometry granular flows.

6.1 Strong Granular Compaction around a Cylinder

In this section, the 2D extension to the granular piston compaction problem, discussed in section 4.4, is analyzed to demonstrate the solver's ability to handle 2 dimensional moving geometries. The piston face was replaced by a cylinder of radius $R = 4$ centered initially at coordinate $[30, 10]$ in a domain of size 40×20 dimensionless units. Square computational cells of size $\Delta x = 0.16$ were used with pass-through boundary conditions used at all edges of the domain. The initial setup and phase properties can be seen in Table 7.

Solid	Value	Gas	Value	Solid EOS	Value	Gas EOS	Value
α_1	0.73	α_2	0.27	γ_1	5	γ_2	1.35
ρ_1	5.0293	ρ_2	2.6470^{-3}	π_1	0.16021	b_2	0.37778
p_1	3.5681×10^{-4}	p_2	1.1843×10^{-5}	C_v^1	0.625	C_v^2	1
u_1	-0.013318	u_2	-0.013318	q_1	0.11795		
v_1	0	v_2	0				

Table 7: Initial conditions and equation of state parameters for the 2D cylinder compaction problem

With reference states: $\alpha_{1,0} = 0.73$; $\alpha_{2,0} = 0.73$; $\rho'_{1,0} = 1900kg/m^3$; $\rho'_{2,0} = 1kg/m^3$; $p'_{1,0} = 7.6$ MPa; $p'_{2,0} = 0.25225$ MPa; $\pi'_{1,0} = 3412.4$ MPa; $b'_{2,0} = 0.001m^3/kg$; $C_v^{1,0'} = 1500J/(kgK)$; $C_v^{2,0'} = 2400J/(kgK)$.

Figure 24 shows the development of the volume fraction over time. Initially at $t = 0$, the volume fraction is 0.73 throughout the domain. Initially the solid volume fraction rapidly increases at the righthand side of the cylinder, whilst it decreases at the left - demonstrating both compaction and cavitation respectively. Initially, the rate of compaction is very high, localised directly adjacent to the cylinder. However, over time, both the rate of compaction slows and becomes increasingly less localized, effectively smoothing the compaction curve. Physically, this can be thought of as the solid grains closest to the cylinder face nearing their maximum packing density and therefore any further compaction would require compression of the solid particles. This compression would require significant amounts of energy and thus it is far more efficient for these compacted particles to compact the particles immediately in their path. Over time, it can also be seen that the velocity profile around the cylinder, shown in Figure 25, becomes more spread out as the material nearest to the cylinder is compacted. The material begins to travel with the cylinder and acts as a solid, further compacting material in its path.

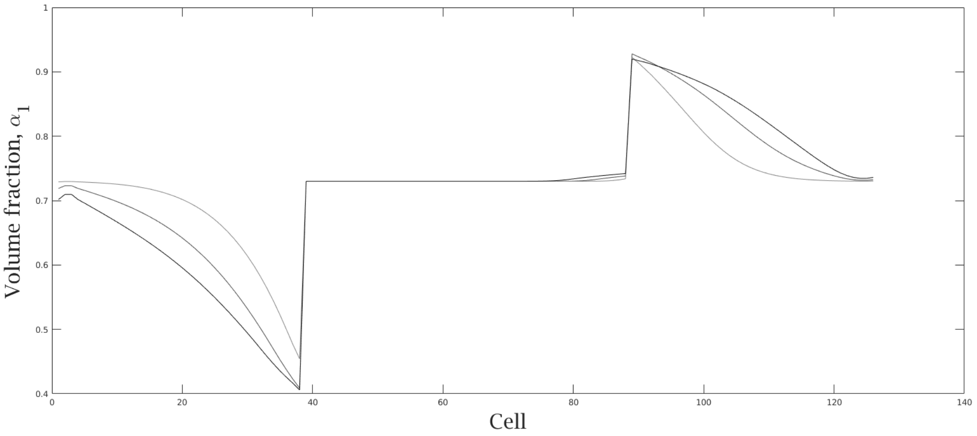


Figure 24: Centerline plots of (a) solid volume fraction and (b) normalized solid density at times $t = 50$ (light grey), $t = 100$ (dark grey) $t = 150$ (black)

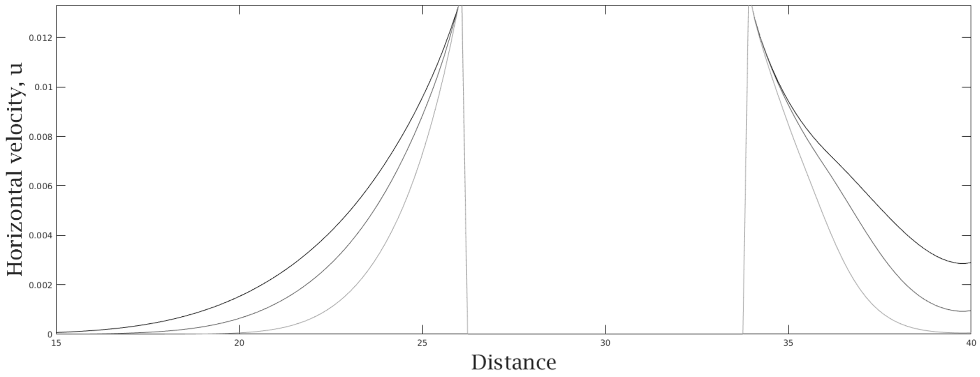


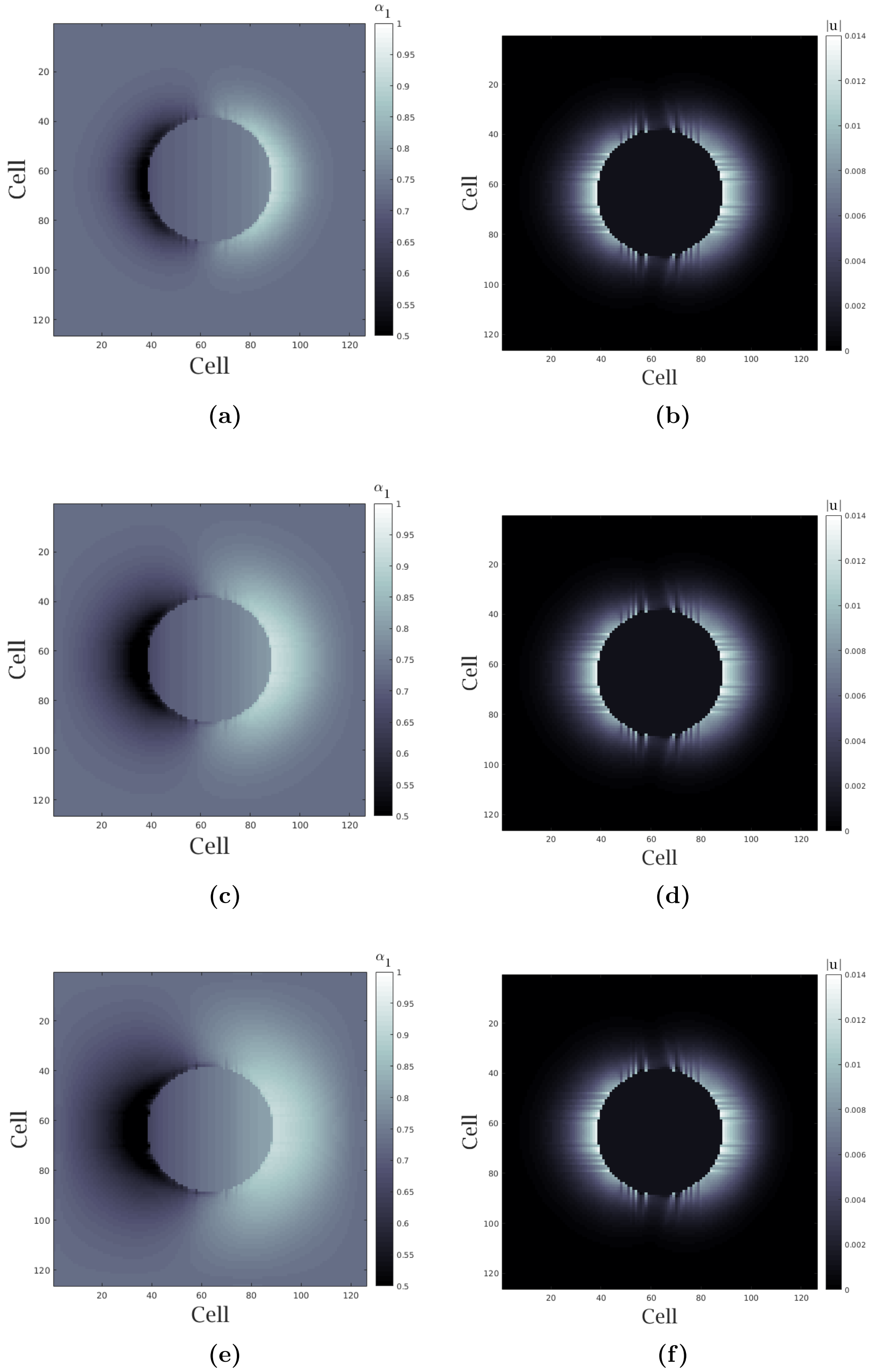
Figure 25: Centerline plots of horizontal velocity at times $t = 50$ (light grey), $t = 100$ (dark grey) $t = 150$ (black) for a moving cylinder.

Two dimensional images showing the development of the solid volume fraction and velocity magnitude (relative to the background flow) are shown in Figure 26. As expected, a region of compaction and cavitation are observed to the right and left of the cylinder, which become thinner and smaller in magnitude toward the top and bottom edges of the cylinder, were the compaction is identical to the background flow - $\alpha = 0.73$. The velocity image shows that initially, the velocity is concentrated in a region directly around the cylinder, before diffusing over time. Spurious oscillations in the velocity magnitude can also be observed, where thin spike-like regions of lower velocity magnitude extend outwards from the cylinder into the background flow. These are caused by the resolution of the cylinder. Numerical discretization of the curved cylinder profile into a Cartesian grid system results in the cylinder shape becoming 'pixelated' creating jagged edges. These sharp corners produce regions of lower velocity, where the pixelated cylinder is perpendicular to the flow and higher velocity, where the pixelated cylinder wall is parallel to the flow.

Self-Consistency Study

A second simulation was run in the frame of reference of the background granular flow with a moving cylinder. This used exactly the same initial conditions, boundary conditions and equation of state parameters as shown in Table 7, however the initial horizontal velocities, u_1 and u_2 were set to zero and the cylinder given a constant velocity of $u_c = 0.013318$. The two solid volume fraction centerline profiles for the fixed cylinder and moving cylinder are shown comparatively in Figure 27.

The comparative plots show a good concordance with one another with low levels of discrepancy between the two, especially for the final plot at $t = 150$. At this final time, the two cylinders are located in the exact same position relative to the grid, meaning their geometries are discretized in the exact same way. This is not always the case. As the cylinder moves within the domain, it is effectively 'sampled' by the ghost fluid method, distorting the discretized shape. This distortion is responsible for the small differences in volume fraction at the cylinder edges in Figures 27(a) and (b).

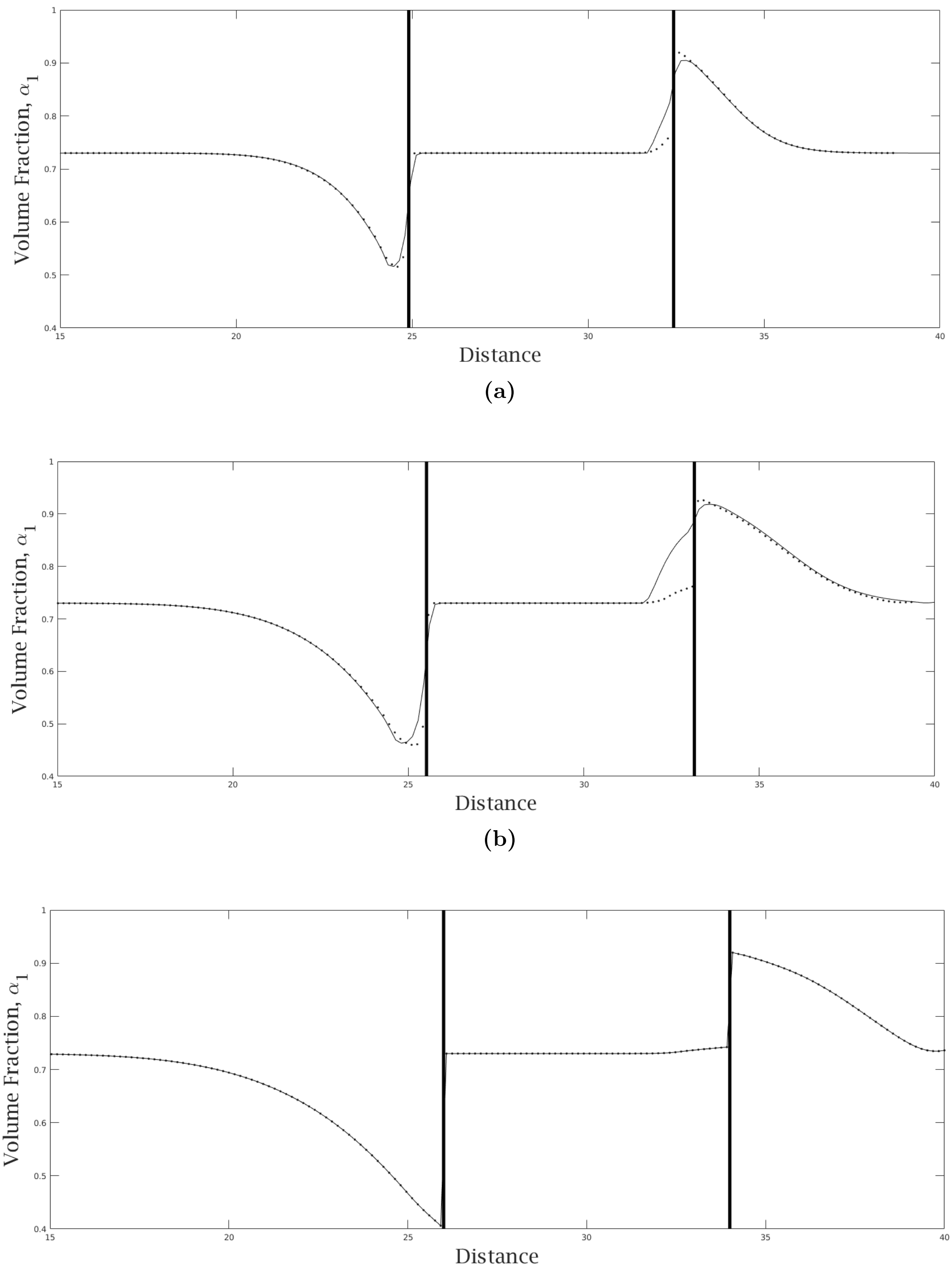


(a) **(b)**

(c) **(d)**

(e) **(f)**

Figure 26: Images showing the evolution of solid volume fraction at times (a) $t = 50$ (c) $t = 100$ (e) $t = 150$ and total density at times (b) $t = 50$ (d) $t = 100$ (f) $t = 150$



(a)

(b)

(c)

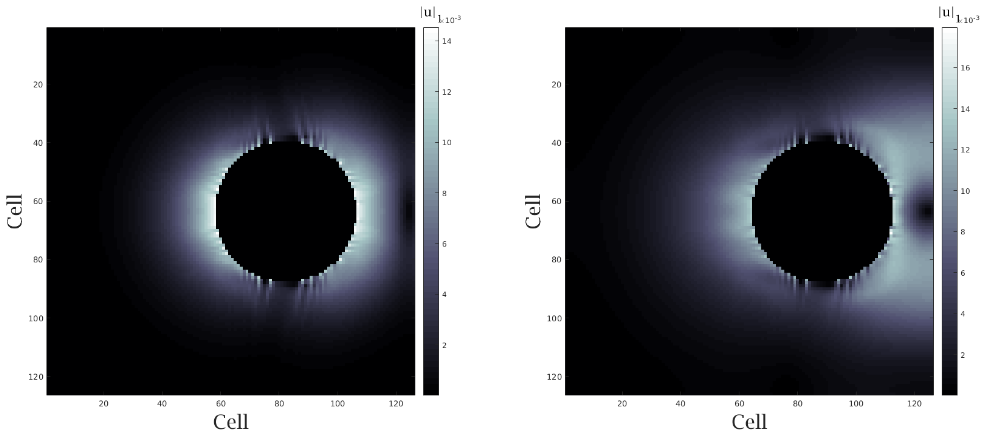
Figure 27: Centerline plots of solid volume fraction against displacement at times (a) $t = 50$ (b) $t = 100$ (c) $t = 150$ using a fixed cylinder (solid line) and a moving cylinder (dotted line). The thick solid lines show the location of the edges of the cylinder.

6.2 Granular Crushing

The aim of this study was to demonstrate the ability of the solver to simulate granular crushing. This is the process in which two solid objects move towards one another, locally increasing solid packing density until it reaches is theoretical maximum. In this case, the same moving cylinder setup described in Section 6.1 is used, but with the right hand 'pass-through' boundary condition replaced by a solid wall.

The cylinder initially starts at coordinates (14,10) in a boundary defined by the rectangle $[0, 20] \times [0, 20]$. At $t = 0$ the cylinder moves towards the righthand solid wall with horizontal velocity $u_{cL} = 0.013318$, compacting the adjacent solid in the process. As seen in Figure 29, up to $t = 50$, the compaction process happens in an identical fashion to the problem in Section 6.1, as the compaction wave has not yet reached the solid wall. Following this, the solid volume fraction between the solid wall and the cylinder increases dramatically, reaching the theoretical maximum of 0.975 predicted in Schwendeman et al. [5].

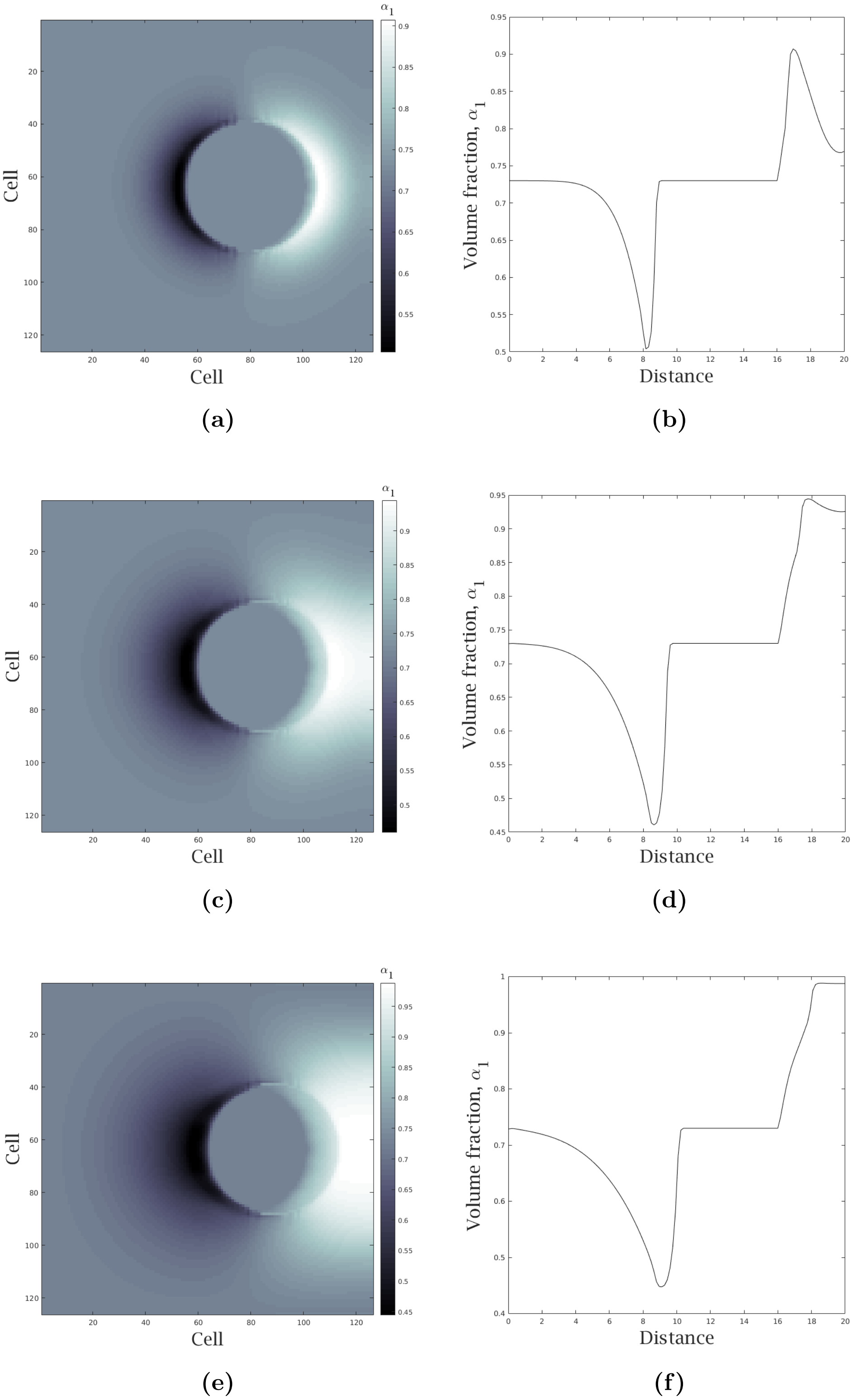
The velocity magnitude profile in Figure 28 shows a significant stagnation in velocity with the zone of maximum compaction, with a higher velocity at the edges of the zone. This increased velocity acts to disperse solid mass and stop compression of the grains within this compaction zone. It should be noted that the stronger compaction effect observed within this model may mean that the Schwendeman model is not applicable in this context and thus the solution depicted here is not a true physical solution. Unfortunately there is very little experimental literature on this particular class of problems and thus this is intended to demonstrate the model capabilities. However a more physically accurate simulation would only require the addition of a more relevant compaction model.



(a)

(b)

Figure 28: Plots of the evolution of velocity magnitude at times (a) $t = 75$ and (b) $t = 150$ for a granular material being crushed between a cylinder and a wall.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 29: Heatmap images of solid volume fraction against displacement at times (a) $t = 50$ (c) $t = 100$ (e) $t = 150$ and their corresponding centerline plots at times (b) $t = 50$, (d) $t = 100$ and (f) $t = 150$ for a granular material being crushed between a cylinder and a wall

7 Discussions

The ADER-WENO MGFM algorithm developed within this work demonstrates a novel capability in modeling granular multiphase flow interactions with moving and complex geometries that are consistent with numerical, experimental and theoretical results. In section 6, various aspects of the overall algorithm are discussed, including potential sources of error, further work to increase the algorithm's capabilities and a conclusion describing the scope of the work performed and its applications.

7.1 Discussion of Error

The ADER-WENO method produced high order of accuracy results that were consistent with the verification problems analyzed. However, as demonstrated in the validation problems in Section 4., there were still some minor sources of error, introduced through mathematical assumptions, numerical discretization and solution algorithms used.

Modeling Errors

Firstly the errors produced by the mathematical model are discussed. The BN equations work extremely well in modeling High-Mach number multiphase flows, as seen in the experimental comparisons. However even within these, higher pressures and lower rates of diffusion can be seen than in the experimental observations. As discussed in Saurel, the BN model ignores boundary layer and viscosity effects, which act to diffuse velocity and pressure. This is unimportant within the context of the inert shock tube problem, but may become a significant source of error when looking at 2D problems, especially problems strongly dependent on peak pressure, such as detonation studies. The exchange terms for the BN equations are also not exact. In most cases the compaction equations are empirically derived and only applicable for a small range of Mach numbers/compaction rates. These compaction parameters are also averaged and ignore the contributions of individual particles.

Discretization Errors

The ADER-WENO's high order of accuracy requires significantly more computational power to resolve a single cell than other comparable methods (such as MUSCL-Hancock, or SLIC schemes). The high computational cost severely restricts the allowable resolution of the mesh, generally to under 100,000 cells for a single core CPU, without significant time needed to simulate each problem ($>$ 24 hours). In future work it may be worth comparing the effects of using a lower order of accuracy scheme with a higher resolution and find a compromise. This could be achieved through use of a second order method combined with a high resolution, refined grid.

The method also has issues resolving shock waves within non-mixed multiphase systems. These are systems in which the two phases are distinct, unmixed and separated by a sharp discontinuity in volume fraction. The ADER-WENO method uses the fastest wave speed in order to calculate the time step used and in multiphase systems with large differences in speed of sound, the less stiff phase is solved at a much lower time-step than if it were in a single phase configuration. This leads to significant smearing of shock wave profiles as the error acts as a diffusive source term.

One additional advantage of using the high order ADER-WENO is its ability to resolve large ratio differences ($\approx 10^5$) in volume fraction between the two phases without becoming unstable. Using a first or second order scheme often leads to significant numerical instability near discontinuities in the volume fraction. These instabilities are caused by smearing of the volume fraction profile, over time, this smearing causes localized hot or cold spots where the pressure is increased due to a non-physical dis-equilibrium. However, increasing the ratio in mass fraction above 10^6 still leads to these numerical instabilities.

The use of source term splitting also introduces errors into the system. In general, however these errors often act in a similar way to the source terms themselves-acting to relax pressure and velocity difference between the two phases. The actual implementation of the splitting is also done inefficiently, using a simplistic trapezium rule integration instead of a Runge-Kutta or backwards Euler method. This was done in order to keep the solution of the source terms simple and thus allow easy and fast modification of the exchange models described in Section 2.2. Not using a more sophisticated solver has limited drawback in this setting, the trapezium rule method is computationally cheap compared to the ADER-WENO and because of the small step size and reasonably non-stiff source terms, introduces little numerical error.

Discretization error is not only introduced by the discretization of the underlying PDE, but also the geometry of the domain. The use of a fixed Cartesian grid, combined with the lower resolution imposed by the ADER-WENO meant that curved geometry become significantly 'pixelated'. This means that a physically smooth geometry will have sharp features in the computational domain, producing some numerical oscillations. In the problems discussed above, these oscillations are mostly small in magnitude and dampen fairly rapidly, however, this may become problematic in problems strongly dependent on pressure, again such as detonation problems where higher pressures can produce premature ignition and detonation.

7.2 Further Work

Some potential improvements to the model are proposed here, with the aim of: reducing error, increasing calculation speed and widening utility.

One of the largest sources of error in the scheme comes from the lower resolution imposition of the domain. This means that the solver is unable to resolve small discontinuities like bubbles or thin vortices, whilst the accuracy of recreating curved geometry is lowered. One of the simplest ways to improve upon this would be to implement an adaptive mesh refinement scheme. The AMR algorithm subdivides the domain into high resolution sections (around sharp, high gradient flow features) and low resolution sections (for smooth, slow moving regions). Initially a coarse grid is defined within the domain, the algorithm then finds coarse cells containing these flow features and creates a secondary mesh within each of these cells, using neighbouring reconstructed values as boundary conditions. This has the dual effect of both reducing processor time and increasing accuracy, as well as better resolving solid geometry features.

Alternatively an unstructured, moving mesh can be used through the ALE formulation of the BN equations. This method can precisely resolves the exact geometry of the domain, but is much more complicated to implement than simple adaptive mesh refinement.

It would also be relatively easy to extend the solver for use in three dimensional problems, allowing modeling of more complicated multiphase granular phenomena, such as river bed formation and geophysical mass flows in 3D topography. However, moving into 3 dimensional geometry would require a significant reduction in the algorithm run time for problems to be tractable in reasonable time. As well using the adaptive mesh refinement described above, algorithm parallelization is also required, something which the ADER-WENO routine is well suited to.

Two methods of parallelization can be implemented, or even combined, GPU parallelization and distributed processor parallelization, with GPU parallelization requiring significant modifications to the code, but providing a much more cost-effective solution. A standard method to parallelize the code across separate cores or CPU's is to split the problem domain up into smaller, overlapping domains. Each of these domains is solved on an independent CPU for one time step, before data from the overlapping cells is shared and used as the boundary conditions for solution in the next time step. As the ADER-WENO only requires N adjacent data points to calculate the reconstruction piecewise and intercell flux (where N is the scheme order). Thus only a small amount of data is transfered between each core, leading to low levels of data latency.

Alternatively, GPU processing can be used. Once the WENO reconstruction has been calculated, communication between cells is not required and thus each GPU core deals with an individual computational cell as intercell communication is only required to generate each Riemann problem and finally perform the update formula. Code was written to test GPU performance in evaluating and inverting flux matrices (the process required to approximately solve the intercell Riemann problem). Whilst there was only a 30% speedup, a large proportion of this time was spent performing data transfer between the GPU and CPU RAM, thus if the entire ADER-WENO was rewritten for solution on the GPU, there would be significant computational speedup.

As well as parallelization, there are other potential ways to increase the speed of the algorithm. Firstly using an semi- analytic DG solver, such as a Newton-Raphson iterator, in place of the split method would rapidly increase both the speed and stability of space-time polynomial generation, especially when paired with a better initial guess generated through, for example, a first order Riemann solver approximation.

8 Conclusion

Within this work, a high resolution numerical model for multiphase granular flows with moving solid boundaries has been presented. Initially, the physics of granular multiphase flows was analyzed, before contrasting the various potential solution methodologies and models. The resulting solution method was chosen to be an arbitrary order ADER-WENO type solver for the Baer-Nunziato equations, combined with a ghost fluid method for solid boundaries. The numerical algorithm was designed and implemented within the C++ programming language using the Armadillo library, before being validated against experimental, numerical and analytical results.

The validation cases demonstrated that the resulting algorithm was robust, numerically accurate and, within a reasonable parameter, physically accurate when paired with an appropriate granular exchange model. The numerical results also displayed many of the key physical phenomena present within granular multiphase flows - namely: compaction; dispersion; cavitation; crushing and shock wave generation. The next paragraphs describe some example applications for this work, that could be implemented in conjunction with some of the extensions outlined within the previous section.

One extremely active area of research is the study of granular explosives. These are a highly common type of explosive, common within military applications as well as mining operations or even fireworks. These rely on having a container filled with a mixture of reactive granules and oxygen, this can then be detonated in a number of ways, such as having a decrease in the container - causing a pressure and temperature increase, or through friction. In many of these applications, the detonation can be strongly affected by the geometry of the problem and thus having an accurate way to model these effects can be highly important for both efficiency and safety considerations. With minimal modification, the algorithm presented within this work could be used to model granular detonation type problems, simply requiring a chemically reactive equation of state along with the mass exchange parameters described in [5].

Additionally the model could be used for fluidization simulations, whereby a granular material is energized from a static state to a dynamic one by a fluid propagating through. This is a common phenomena observed within a wide range of scenarios. Within the natural world, these flows are present in sand volcanoes and fluid escape structures, whereby a fluid or granular material is forced through a secondary granular material. Industrial examples of these flows include fluidic catalytic cracking within the oil production process or purification of raw chemicals.[41]

In conclusion, the numerical algorithm presented here has the capability to produce high resolution, novel results that are both numerically stable and physical. With minimal modification, the work presented here also has the capability to be integrated into a wider range of numerical models that have strong applications in both industry and environmental work.

9 References

[1] D. Chehata, R. Zenit, and C. Wassgren, "Dense granular flow around an immersed cylinder," *Physics of fluids*, vol. 15, no. 6, pp. 1622–1631, 2003.

[2] R. Saurel and R. Abgrall, "A multiphase godunov method for compressible multifluid and multiphase flows," *Journal of Computational Physics*, vol. 150, no. 2, pp. 425 – 467, 1999.

[3] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction.* Springer Science & Business Media, 2013.

[4] M. Dumbser, O. Zanotti, A. Hidalgo, and D. S. Balsara, "Ader-weno finite volume schemes with space–time adaptive mesh refinement," *Journal of Computational Physics*, vol. 248, pp. 257–286, 2013.

[5] D. Schwendeman, C. Wahle, and A. Kapila, "A study of detonation evolution and structure for a model of compressible two-phase reactive flow," *Combustion Theory and Modelling*, vol. 12, no. 1, pp. 159–204, 2008.

[6] G. Falcone, G. Hewitt, C. Alimonti, B. Harrison, *et al.*, "Multiphase flow metering: current trends and future developments," in *SPE annual technical conference and exhibition*, Society of Petroleum Engineers, 2001.

[7] N. I. Kolev and N. Kolev, *Multiphase flow dynamics*, vol. 1. Springer, 2005.

[8] M. P. Duduković, F. Larachi, and P. L. Mills, "Multiphase catalytic reactors: a perspective on current knowledge and future trends," *Catalysis reviews*, vol. 44, no. 1, pp. 123–246, 2002.

[9] H. Udaykumar, H.-C. Kan, W. Shyy, and R. Tran-Son-Tay, "Multiphase dynamics in arbitrary geometries on fixed cartesian grids," *Journal of Computational Physics*, vol. 137, no. 2, pp. 366 – 405, 1997.

[10] H. H. Hu, N. Patankar, and M. Zhu, "Direct numerical simulations of fluid–solid systems using the arbitrary lagrangian–eulerian technique," *Journal of Computational Physics*, vol. 169, no. 2, pp. 427 – 462, 2001.

[11] C. Wassgren, J. Cordova, R. Zenit, and A. Karion, "Dilute granular flow around an immersed cylinder," *Physics of fluids*, vol. 15, no. 11, pp. 3318–3330, 2003.

[12] P. Jop, Y. Forterre, and O. Pouliquen, "A constitutive law for dense granular flows," *arXiv preprint cond-mat/0612110*, 2006.

[13] R. M. Iverson, "The physics of debris flows," *Reviews of geophysics*, vol. 35, no. 3, pp. 245–296, 1997.

[14] I. Goldhirsch, "Rapid granular flows," *Annual review of fluid mechanics*, vol. 35, no. 1, pp. 267–293, 2003.

[15] M. Baer and J. Nunziato, “A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials,” *International journal of multiphase flow*, vol. 12, no. 6, pp. 861–889, 1986.

[16] J. Trepanier, M. Reggio, H. Zhang, and R. Camarero, “A finite-volume method for the euler equations on arbitrary lagrangian-eulerian grids,” *Computers & fluids*, vol. 20, no. 4, pp. 399–409, 1991.

[17] W. Boscheri and M. Dumbser, “A direct arbitrary-lagrangian–eulerian ader-weno finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3d,” *Journal of Computational Physics*, vol. 275, pp. 484–523, 2014.

[18] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, “A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method),” *Journal of computational physics*, vol. 152, no. 2, pp. 457–492, 1999.

[19] S. K. Sambasivan and H. UdayKumar, “Ghost fluid method for strong shock interactions part 2: Immersed solid boundaries,” *AIAA J*, vol. 47, no. 12, pp. 2923–2937, 2009.

[20] C. W. Hirt and B. D. Nichols, “Volume of fluid (vof) method for the dynamics of free boundaries,” *Journal of computational physics*, vol. 39, no. 1, pp. 201–225, 1981.

[21] D. M. Anderson, G. B. McFadden, and A. A. Wheeler, “Diffuse-interface methods in fluid mechanics,” *Annual review of fluid mechanics*, vol. 30, no. 1, pp. 139–165, 1998.

[22] G. Hou, J. Wang, and A. Layton, “Numerical methods for fluid-structure interaction — a review,” *Communications in Computational Physics*, vol. 12, no. 2, p. 337–377, 2012.

[23] T. Belytschko, “Fluid-structure interaction,” *Computers & Structures*, vol. 12, no. 4, pp. 459–469, 1980.

[24] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, “A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method),” *Journal of Computational Physics*, vol. 152, no. 2, pp. 457 – 492, 1999.

[25] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction.* Springer Science & Business Media, 2013.

[26] B. Van Leer, “Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov’s method,” *Journal of computational Physics*, vol. 32, no. 1, pp. 101–136, 1979.

[27] C. Parés, “Numerical methods for nonconservative hyperbolic systems: a theoretical framework.,” *SIAM Journal on Numerical Analysis*, vol. 44, no. 1, pp. 300–321, 2006.

[28] S. Tokareva and E. F. Toro, "Hllc-type riemann solver for the baer–nunziato equations of compressible two-phase flow," *Journal of Computational Physics*, vol. 229, no. 10, pp. 3573–3604, 2010.

[29] M. Dumbser and W. Boscheri, "High-order unstructured lagrangian one-step weno finite volume schemes for non-conservative hyperbolic systems: applications to compressible multi-phase flows," *Computers & Fluids*, vol. 86, pp. 405–432, 2013.

[30] X. Rogue, G. Rodriguez, J. Haas, and R. Saurel, "Experimental and numerical investigation of the shock-induced fluidization of a particles bed," *Shock Waves*, vol. 8, no. 1, pp. 29–45, 1998.

[31] R. Menikoff, "Empirical equations of state for solids," in *ShockWave Science and Technology Reference Library*, pp. 143–188, Springer, 2007.

[32] G. K. Batchelor, *An introduction to fluid dynamics.* Cambridge university press, 2000.

[33] O. Zanotti and M. Dumbser, "Efficient conservative ader schemes based on weno reconstruction and space-time predictor in primitive variables," *Computational Astrophysics and Cosmology*, vol. 3, no. 1, p. 1, 2016.

[34] R. P. Fedkiw, "Coupling an eulerian fluid calculation to a lagrangian solid calculation with the ghost fluid method," *Journal of Computational Physics*, vol. 175, no. 1, pp. 200–224, 2002.

[35] E. Kreyszig, *Advanced engineering mathematics.* John Wiley & Sons, 2007.

[36] B. Van Leer, "On the relation between the upwind-differencing schemes of godunov, engquist—osher and roe," in *Upwind and high-Resolution schemes*, pp. 33–52, Springer, 1997.

[37] B. Van Leer, "Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method," *Journal of computational Physics*, vol. 32, no. 1, pp. 101–136, 1979.

[38] C. Sanderson, "Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments," 2010.

[39] M. Dumbser, A. Hidalgo, and O. Zanotti, "High order space–time adaptive ader-weno finite volume schemes for non-conservative hyperbolic systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 359–387, 2014.

[40] T. Liu, B. Khoo, and K. Yeo, "Ghost fluid method for strong shock impacting on material interface," *Journal of Computational Physics*, vol. 190, no. 2, pp. 651–681, 2003.

[41] D. Geldart, "Types of gas fluidization," *Powder technology*, vol. 7, no. 5, pp. 285–292, 1973.